



Frederico Osvaldo Martins

Licenciado em Engenharia Eletrotécnica e Computadores

Seguidor solar de dois eixos com motores passo-a-passo

Dissertação para obtenção do Grau de Mestre em
Engenharia de Energias Renováveis

Orientador: Professor Doutor Pedro Miguel Ribeiro Pereira,
Professor Auxiliar, FCT-UNL

Coorientador: Professor Doutor João Francisco Alves Martins,
Professor Auxiliar, FCT-UNL

Júri:

Presidente: Doutor Rui Manuel Leitão Santos Tavares

Arguente: Doutor João Miguel Murta Pina

Vogal: Doutor Pedro Miguel Ribeiro Pereira



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Setembro 2016

Seguidor solar de dois eixos com motores passo-a-passo

Copyright © Frederico Osvaldo Martins, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa

“A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.”

Agradecimentos

Ao Professor Pedro Miguel Ribeiro Pereira pela sua orientação, rigor científico, partilha de conhecimentos e disponibilidade durante a realização deste trabalho. Ao Professor João Francisco Alves Martins, pela sua dedicação como coorientador. Aos meus pais, pelo incentivo e apoio não só durante a realização deste projeto mas também por todo o apoio que me têm dado ao longo da vida. Aos meus irmãos e amigos mais chegados pela amizade e companheirismo de longa data. Ao responsável e à equipa de trabalho da empresa Técnicos de Máquinas Reunidos, por terem tornado possível a construção de uma engrenagem necessária para o desenvolvimento deste trabalho. A todos, muito obrigado.

Resumo

Os seguidores solares, além de servirem de suporte para os painéis solares, seguem o movimento do sol ao longo do dia e procuram manter o plano do painel perpendicular aos raios de sol, aumentando assim o desempenho do painel solar. Este trabalho tem como objetivo a análise do desempenho de um seguidor solar efetuando-se a distinção entre o desempenho do seguidor com um sistema de seguimento baseado em sensores de luminosidade e um sistema de seguimento horário sem sensores de luminosidade. Pretende-se também analisar o desempenho de um seguidor solar de dois eixos e comparar os resultados em relação ao seguimento apenas da elevação, do azimute do sol, e de um painel instalado num sistema fixo. Para alcançar esses objetivos, desenvolveu-se e implementou-se um seguidor de dois eixos com motores passo-a-passo, sensores de luminosidade e algoritmos de posicionamento solar, permitindo assim simular os cenários especificados e analisar os resultados. O desempenho que se obteve poderia ser maior, em parte porque não foi possível aproveitar o sol totalmente durante a manhã e ao final da tarde por causa de obstáculos. Ainda assim, foi possível concluir que os seguidores solares de dois eixos aumentam o desempenho do painel solar fotovoltaico de forma aproximadamente linear ao longo do dia; o seguimento apenas em azimute aumenta mais o desempenho ao início e ao final do dia; e o seguimento só em elevação aumenta mais o desempenho a meio do dia. Também se concluiu que o desempenho do seguimento com sensores de luminosidade foi maior do que o seguimento horário sem sensores de luminosidade.

Termos chave: Algoritmos de posicionamento solar, Arduino, seguidor solar de dois eixos, seguidor solar de um eixo, sensor LDR.

Abstract

A solar tracker, besides of serving as a support for the photovoltaic panel, tracks the movement of the sun throughout the day maintaining the profile of the panel perpendicular to the sun rays, thus increasing the performance of a photovoltaic panel. The objective of this study is to analyze the performance of a time based solar tracker in relation to a solar tracker with light sensors. It also aims to analyze the performance of a double axis solar tracker, a one-axis solar tracker and a fixed system. To achieve those objectives, a two axis solar tracker was developed with step-by-step motors, light sensors and solar positioning algorithms, enabling the simulation of the specified scenarios and the analysis of the results. The achieved performance could be higher, in part because it was not possible to make a full use of the sun during the morning and late afternoon because of obstacles. Still, it was possible to conclude that two axes solar trackers increase the performance of solar panels throughout the day in a linear manner; tracking only in azimuth increases the performance further at the beginning and the end of the day; and tracking only in elevation increases the performance further in the middle of the day. It was also concluded that the performance is greater with a tracking system based on light sensors instead of a time based solar tacker without light sensors.

Keywords: Arduino, LDR sensor, one axis solar tracker, solar positioning algorithms, two axis solar tracker.

Índice

1. INTRODUÇÃO	1
1.1 CONTEXTO E MOTIVAÇÕES	1
1.2 OBJETIVO	2
1.3 ESTRUTURA DA DISSERTAÇÃO.....	3
2. ESTADO DA ARTE	5
2.1 TIPOS DE SEGUIDORES SOLARES	6
2.1.1 Seguidor solar de um eixo.....	6
2.1.2 Seguidor solar de dois eixos.....	7
2.2 VANTAGENS E DESVANTAGENS	7
2.3 DESEMPENHO DOS SEGUIDORES SOLARES	8
3. COMPONENTES DO SISTEMA DO SEGUIDOR SOLAR	13
3.1 SISTEMA MECÂNICO	14
3.2 MOTORES ELÉTRICOS.....	16
3.3 SENSORES LDR	16
3.4 UNIDADE DE CONTROLO	18
3.5 ALGORITMOS DE POSICIONAMENTO SOLAR	19
4. IMPLEMENTAÇÃO DO SEGUIDOR SOLAR	21
4.1 MOTOR PASSO-A-PASSO PD3-109-57	22
4.2 CONTROLADOR ARDUINO UNO	24
4.3 MÓDULO SOLAR FOTOVOLTAICO.....	26

4.4	SENSORES DE LUMINOSIDADE	28
4.5	ALGORITMOS DE POSICIONAMENTO SOLAR	30
4.6	SISTEMA DE BLOQUEIO AUTOMÁTICO.....	30
4.7	CAIXA DO SEGUIDOR SOLAR.....	31
5.	RESULTADOS E DISCUSSÃO	33
6.	CONCLUSÃO	39
	BIBLIOGRAFIA	41
	ANEXOS	45
	ANEXO 1: MEDIÇÕES E CÁLCULOS EFETUADOS	45
	ANEXO 2: CÓDIGO DO PD3-109-57 (MOTOR 1)	46
	ANEXO 3: CÓDIGO DO PD3-109-57 (MOTOR 2)	47
	ANEXO 4: CÓDIGO DO ARDUINO (SEGUIDOR COM SENSORES LDR).....	48
	ANEXO 5: CÓDIGO DO ARDUINO (SEGUIDOR COM ALGORITMO PSA)	55

Índice de figuras

FIGURA 1.1 - ELEVAÇÃO SOLAR AO LONGO DO ANO, ENTRE A LATITUDE NORTE E O EQUADOR.....	2
FIGURA 1.2 - ÂNGULO DE ELEVAÇÃO, DE ZÉNITE E DE AZIMUTE.....	2
FIGURA 2.1 – ESTIMATIVA DO CRESCIMENTO GLOBAL DO MERCADO DOS SEGUIDORES SOLARES.....	5
FIGURA 2.2 – SEGUIDOR DE UM EIXO DO TIPO POLAR	6
FIGURA 2.3 - SEGUIDOR DE UM EIXO DO TIPO HORIZONTAL	6
FIGURA 2.4 - SEGUIDOR SOLAR DE DOIS EIXOS	7
FIGURA 2.5 – DESEMPENHO DE UM SEGUIDOR SOLAR DE DOIS EIXOS EM COMPARAÇÃO COM UM SISTEMA FIXO	8
FIGURA 2.6 - POTÊNCIA DO SEGUIDOR DE UM EIXO E DE UM SISTEMA COM O PAINEL FIXO	9
FIGURA 2.7 - POTÊNCIA DO SEGUIDOR DE DOIS EIXOS E DE UM SISTEMA COM O PAINEL FIXO	9
FIGURA 2.8 - DISPOSIÇÃO DOS SENSORES LDR.....	10
FIGURA 2.9 - POTÊNCIA DO SEGUIDOR SOLAR E DE UM PAINEL FIXO NUM DIA SEM NUVENS.....	11
FIGURA 2.10 - POTÊNCIA DO SEGUIDOR SOLAR E DE UM PAINEL FIXO NUM DIA COM NUVENS	11
FIGURA 3.1 - UNIDADE DE ROTAÇÃO DE UM EIXO	14
FIGURA 3.2 – ILUSTRAÇÃO DO FUNCIONAMENTO DA UNIDADE DE ROTAÇÃO	14
FIGURA 3.3 - UNIDADE DE ROTAÇÃO DE DUPLO EIXO DESENVOLVIDA PELA KINEMATICS MANUFACTURING, INC	15
FIGURA 3.4 - UNIDADE DE ROTAÇÃO COM RODAS DENTADAS DESENVOLVIDO PELA IMO.....	15
FIGURA 3.5 - SISTEMA COM RODAS DENTADAS	15
FIGURA 3.6 - ATUADOR LINEAR DESENVOLVIDO PELA ULTRAMOTION	16
FIGURA 3.7 – TIPOS DE AGRUPAMENTO COM SENSORES LDR NUM SEGUIDOR DE DOIS EIXOS.....	17
FIGURA 3.8 - EXEMPLO DA ORGANIZAÇÃO DOS SENSORES LDR (CdS) NOS SEGUIDORES SOLARES	17
FIGURA 3.9 - VETOR E ÂNGULOS SOLARES COM ORIGEM NA LOCALIZAÇÃO Q	19
FIGURA 4.1 - MOTOR PASSO-A-PASSO PD3-109-57.....	21

FIGURA 4.2 - DIAGRAMA DE BLOCOS DO SEGUIDOR SOLAR	22
FIGURA 4.3 – TERMINAL GPI1, GPI2 E DISABLE DO PD3-109-57	22
FIGURA 4.4 - DIAGRAMA DE BLOCOS DO MICROCONTROLADOR PD3-109-57	23
FIGURA 4.5 – DIAGRAMA DE BLOCOS DO PD3-109-57	23
FIGURA 4.6 - ALGORITMO GLOBAL DO ARDUINO	24
FIGURA 4.7 - ALGORITMO ARDUINO: SUB-ROTINA SEGUIR ELEVAÇÃO	25
FIGURA 4.8 - ALGORITMO ARDUINO: SUB-ROTINA SEGUIR AZIMUTE	25
FIGURA 4.9 – DESATIVAÇÃO DO MOTOR ATRAVÉS DO ARDUINO	26
FIGURA 4.10 - SEGUIDOR SOLAR DESENVOLVIDO COM O MÓDULO BANGKOK SOLAR	26
FIGURA 4.11 - CURVA I-V E POTÊNCIA, REFERENTE À TABELA 4.2.....	27
FIGURA 4.12 - CURVA I-V DE UMA CÉLULA FOTOVOLTAICA DE SILICONE	28
FIGURA 4.13 - DISTRIBUIÇÃO DOS SENSORES LDR.....	28
FIGURA 4.14 - DIVISOR DE TENSÃO COM LDR.....	29
FIGURA 4.15 - SELENOIDE CILÍNDRICO 24 VDC PARA O BLOQUEIO DO AZIMUTE	30
FIGURA 4.16 - CONSTITUINTES INTERNOS DA CAIXA DO SEGUIDOR SOLAR.....	31
FIGURA 5.1 – CONVERSÃO ENERGÉTICA DO MÓDULO NO SEGUIDOR SOLAR COM SENSORES LDR	34
FIGURA 5.2 – CONVERSÃO ENERGÉTICA DO MÓDULO NO SEGUIDOR SOLAR COM O ALGORITMO PSA.....	34
FIGURA 5.3 – ACRÉSCIMO DE POTÊNCIA DO SEGUIDOR SOLAR PSA DE 2 EIXOS EM RELAÇÃO A UM SISTEMA FIXO	36
FIGURA 5.4 – ÂNGULO DE ELEVAÇÃO SOLAR OBTIDO COM O ALGORITMO PSA	37
FIGURA 5.5 – ÂNGULO DE AZIMUTE SOLAR OBTIDO COM O ALGORITMO PSA.....	37
FIGURA 5.6 - VARIAÇÃO DO AZIMUTE E DA ELEVAÇÃO COM O SEGUIDOR VIRADO INICIALMENTE PARA SUL	38

Índice de tabelas

TABELA 4.1 - VOC E ISC ENSAIADO E ESPECIFICADO PARA O MÓDULO KYOCERA KC40	27
TABELA 4.2 -TENSÃO, CORRENTE E POTÊNCIA COM A VARIAÇÃO DA RESISTÊNCIA	27
TABELA 4.3 - TESTE DA RESISTÊNCIA DO LDR EM CENÁRIOS DIFERENTES DE LUMINOSIDADE	29
TABELA 5.1 - CONDIÇÕES DE TESTE DO SEGUIDOR SOLAR COM SENSORES E ALGORITMOS SPA.....	33
TABELA 5.2 - COMPARAÇÃO DO DESEMPENHO DO PAINEL DO SEGUIDOR SOLAR EM RELAÇÃO A UM SISTEMA FIXO	35
TABELA 5.3 - SIMULAÇÃO DOS RESULTADOS DO ALGORITMO PSA	36

Lista de siglas

AC	<i>Alternating Current</i>
BLDC	<i>Brushless DC motors</i>
DC	<i>Direct Current</i>
GPS	<i>Global Positioning System</i>
I_{sc}	<i>Short-circuit current</i>
LDR	<i>Light Dependent Resistor</i>
M_{pp}	<i>Maximum Power Point</i>
NREL	<i>National Renewable Energy Laboratory</i>
PLC	<i>Programable Logic Controller</i>
PSA	<i>Plataforma Solar de Almería</i>
RTC	<i>Real Time Clock</i>
SPA	<i>Solar Position Algorithm</i>
STC	<i>Standard Test Conditions</i>
TMCL	<i>Trinamic Motion Control Language</i>
V_{oc}	<i>Open-circuit voltage</i>

1. Introdução

Neste capítulo introdutório são apresentados os assuntos que serviram de incentivo à elaboração da dissertação, os objetivos, e a estrutura do trabalho.

1.1 Contexto e Motivações

As necessidades energéticas têm aumentado com o desenvolvimento industrial, com o aumento da população mundial e da qualidade de vida. Esta situação torna-se problemática quando a principal fonte de energia para satisfazer essas necessidades provém de combustíveis fósseis tendo em consideração a degradação ambiental causada pela exploração em larga escala deste recurso. O aquecimento global, a poluição do ar e as chuvas ácidas, incitam a exploração de fontes de energia renováveis como a energia solar [1].

Os módulos fotovoltaicos são utilizados para converter a radiação solar em energia elétrica e têm uma eficiência na ordem dos 20% em condições STC [2]. Existem várias tecnologias que podem otimizar essa eficiência como o melhoramento da capacidade de captura de luz ou a otimização da geometria das células solares e a utilização de sistemas de seguimento solar [1], sendo esse último o tema de estudo deste trabalho.

O sol aparenta deslocar-se com uma velocidade constante de aproximadamente 15° por hora ao longo do dia e o seu posicionamento pode ser determinado com recurso ao ângulo de elevação, ângulo de zénite e de azimute [3]. O ângulo de elevação corresponde ao ângulo entre o sol e o horizonte. Este ângulo é variável ao longo do dia, das estações do ano, e da latitude (figura 1.1) [4].

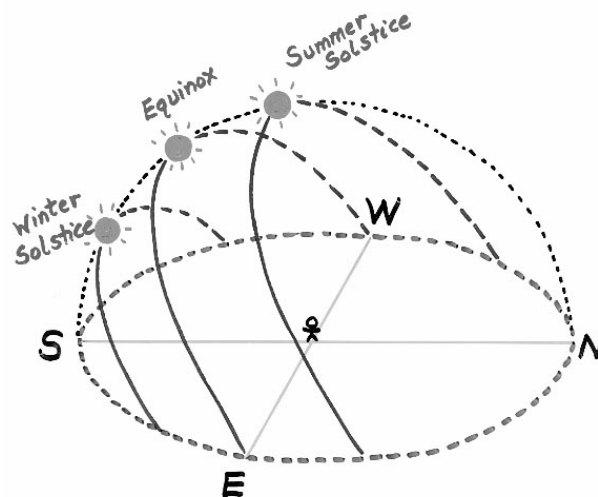


Figura 1.1 - Elevação solar ao longo do ano, entre a latitude norte e o equador
Adaptado de [4]

O ângulo de Zénite representa o angulo entre a vertical e o posicionamento do sol. A direção do sol no plano horizontal é representada pelo ângulo de azimute, como mostra a figura 1.2.

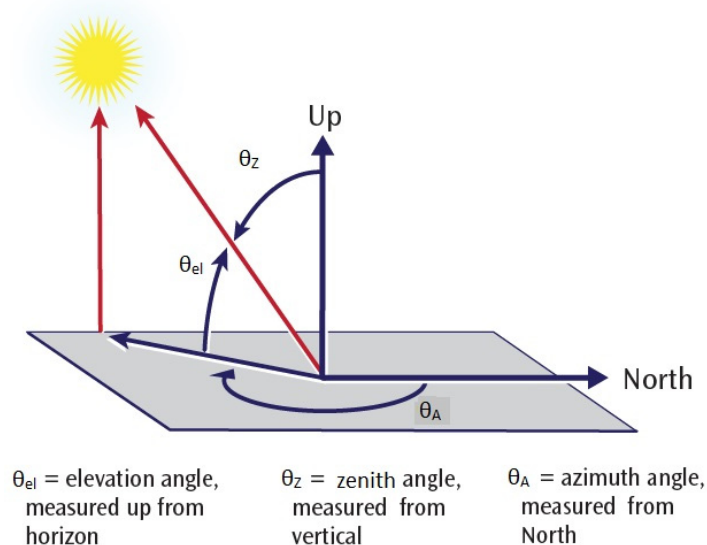


Figura 1.2 - Ângulo de elevação, de zénite e de azimute [5]

1.2 Objetivo

Este trabalho tem como objetivo a análise do desempenho de seguidores solares que se baseiam em sistemas de seguimento com sensores de luminosidade e em sistemas de seguimento horário, sem sensores de luminosidade. Pretende-se também analisar o desempenho de um seguidor solar de dois eixos nas seguintes situações:

- Seguimento com dois eixos ativo;
- Seguimento com um eixo ativo:
 - Variação apenas em azimute;
 - Variação apenas em inclinação.
- Sistema com o painel solar fixo.

Para alcançar esses objetivos, analisaram-se as soluções existentes e desenvolveu-se um seguidor solar de dois eixos com motores passo-a-passo, sensores de luminosidade e algoritmos de posicionamento solar, permitindo assim ensaiar os diferentes cenários e analisar os resultados das medições. Espera-se que este trabalho permita auxiliar a escolha do tipo de seguidor tendo em consideração as características e o desempenho do mesmo.

1.3 Estrutura da Dissertação

A presente dissertação encontra-se dividida em seis capítulos. O Capítulo 1 visa enquadrar o trabalho apresentando as motivações que impulsionaram a elaboração desta dissertação bem como os principais objetivos. O Capítulo 2 apresenta uma breve revisão do estado da arte dos seguidores solares, aborda seguidores solares que existem no mercado e o desempenho dos mesmos. O Capítulo 3 descreve os componentes que tipicamente são considerados no desenvolvimento de seguidores solares. A implementação que foi realizada é apresentada no Capítulo 4 sendo os resultados apresentados no Capítulo 5. Por último, o Capítulo 6 apresenta as principais conclusões do trabalho e sugestões para desenvolvimentos futuros.

2. Estado da Arte

A procura por seguidores solares tem aumentado com o aumento da procura por módulos fotovoltaicos por causa do aumento dos custos da eletricidade e do esgotamento de fontes não renováveis [6]. No período de 2014, a procura por seguidores solares foi maior na Europa, com uma cota de mercado de 38% condicionada maioritariamente pela Itália, Espanha, Alemanha e Grécia. A Europa foi seguida pela América do Norte com uma maior procura nos Estados Unidos. [6]. A figura 2.1 mostra o crescimento global do mercado dos seguidores solares previsto até 2020.

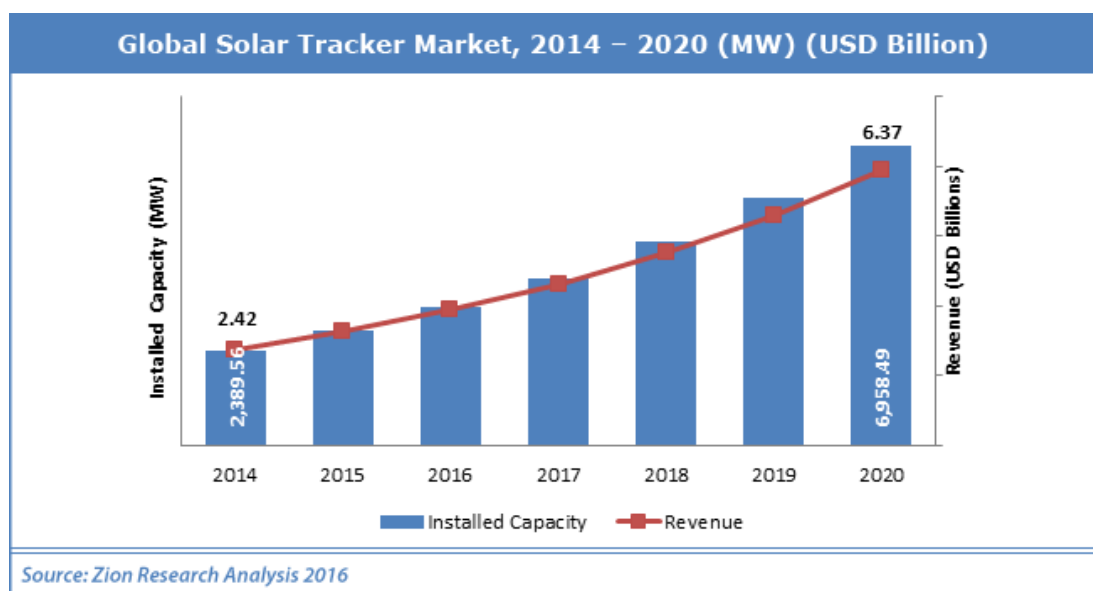


Figura 2.1 – Estimativa do crescimento global do mercado dos seguidores solares [6]

2.1 Tipos de seguidores solares

Os seguidores solares podem ser de um eixo ou de dois eixos e a escolha destes sistemas pode trazer vantagens ou desvantagens como será apresentado adiante neste trabalho. A seguir são apresentadas as características dos seguidores solares de um eixo e de dois eixos.

2.1.1 Seguidor solar de um eixo

Nestes seguidores, o mecanismo roda apenas num plano. O eixo de rotação pode ser polar, quando implementado com uma determinada inclinação, ou horizontal, conforme mostram as figuras 2.2 e 2.3.



Figura 2.2 – Seguidor de um eixo do tipo polar [7]

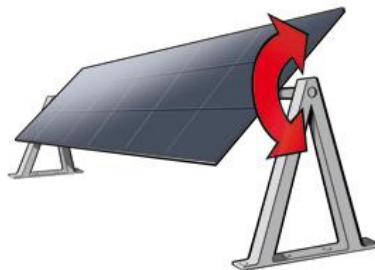


Figura 2.3 - Seguidor de um eixo do tipo horizontal [7]

Os seguidores de um eixo do tipo horizontal são mais adequados em baixas latitudes como nos trópicos e perto do equador e são também estruturalmente mais rígidos e estáveis sendo mais resistentes a danos provocados por tempestades [7]. Os de eixo polar são mais adequados em latitudes altas, afastadas do equador e dos trópicos [7]. A energia anual obtida com a utilização de um seguidor solar de um eixo aumenta aproximadamente até 30% em comparação com um sistema imóvel [8].

2.1.2 Seguidor solar de dois eixos

Os dois eixos de rotação permitem que o seguidor alinhe as células solares para ficarem perpendiculares aos raios de sol ao longo do dia e das estações do ano [7]. A energia anual obtida com a utilização de um seguidor solar de dois eixos (figura 2.4), em comparação com um sistema fixo, aumenta aproximadamente até 40%. Os seguidores de dois eixos acompanham com mais exatidão a gama de movimentos do sol, no entanto, são mais complexos e mais caros [7].



Figura 2.4 - Seguidor solar de dois eixos [7]

2.2 Vantagens e desvantagens

A procura por seguidores solares tem aumentado conforme mencionado no início deste capítulo. Estes sistemas oferecem vantagens e desvantagens em relação a um sistema com o painel fixo, sendo essas apresentadas a seguir.

Vantagens

- Adaptabilidade: existem tipos diferentes de seguidores solares, como de um eixo e de dois eixos. A escolha do tipo de seguidor poderá ser mais adequada para um determinado local, considerando o espaço para instalação, as condições climáticas e a latitude. [9]
- Otimização do espaço: o aumento da produção de energia traduz-se numa otimização do espaço disponível para instalação do sistema, ou seja, os seguidores solares necessitam de menos espaço do que os sistemas fixos para gerar a mesma energia. [9]
- Durabilidade: os avanços tecnológicos da eletrônica e da mecânica reduziram drasticamente a necessidade de manutenção dos seguidores solares [9].
- Aumento do desempenho: os seguidores permitem aumentar a produção de energia até 40% em relação aos sistemas com painéis fotovoltaicos fixos.

Desvantagens

- O custo é maior, aliado à complexidade da tecnologia [9]
- Apesar dos avanços tecnológicos em termos de robustez, é necessária uma maior manutenção em comparação com os sistemas fixos [9]
- Os seguidores solares tipicamente são desenhados para funcionarem em climas com pouca ou nenhuma neve, o que os torna uma solução mais viável em climas mais quentes. Em comparação, os painéis solares fixos suportam condições ambientais adversas com uma maior facilidade do que os seguidores solares [9].

2.3 Desempenho dos seguidores solares

Na Universidade da Síria, desenvolveu-se um seguidor solar fotovoltaico de dois eixos e ensaiou-se o desempenho nos dias 10, 11 e 12 Maio de 2013 (latitude 33.33° N, longitude 36.18° E). O seguidor solar foi construído com quatro sensores LDR, um PLC, dois interruptores fim de curso, dois servomotores DC e a conversão da energia aumentou 30% a 40% em comparação com um painel solar fixo inclinado a 45°. Observou-se também que o desempenho do seguidor solar foi maior ao início e ao final do dia conforme mostra a figura 2.5 [10].

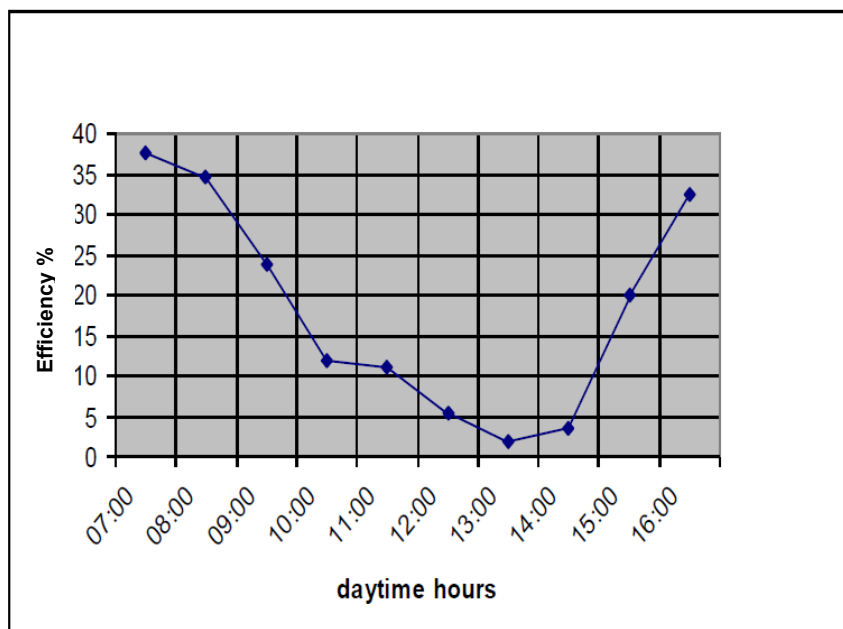


Figura 2.5 – Desempenho de um seguidor solar de dois eixos em comparação com um sistema fixo [10]

A Faculdade de Engenharia Elétrica e Eletrônica de Bangladesh também desenvolveu um seguidor de dois eixos e comparou-se o desempenho com um painel fixo. O seguidor foi construído com cinco sensores LDR, um microcontrolador e dois motores passo-a-passo. Foram

utilizados dois pares de sensores LDR para direcionar o seguidor no plano vertical e horizontal respectivamente. O quinto sensor é responsável pela determinação do modo noturno que coloca o seguidor na posição inicial, virado para Este, e desativa os motores. As medições foram efetuadas a partir das 08:00 até às 17:00 em intervalos de uma hora e o seguidor de dois eixos obteve aproximadamente um aumento de desempenho de 50% em relação a um painel fixo [11].

Na Universidade da Índia adicionou-se ao estudo o seguidor de um eixo. Foram utilizados quatro sensores LDR para o seguimento em dois eixos, e dois sensores LDR para o seguidor de um eixo. Os testes decorreram das 08:00 às 18:00 e as medições foram efetuadas em intervalos de uma hora. As figuras 2.6 e 2.7 apresentam graficamente a potência dos seguidores em comparação com a potência de um painel fixo [12].

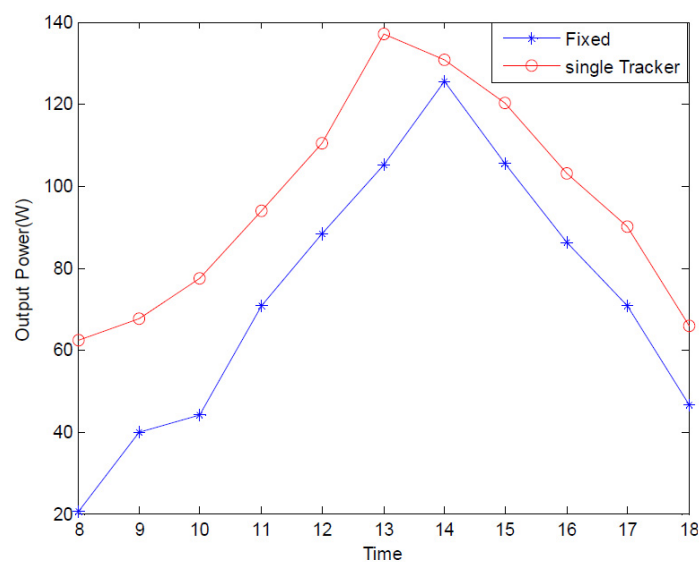


Figura 2.6 - Potência do seguidor de um eixo e de um sistema com o painel fixo [12]

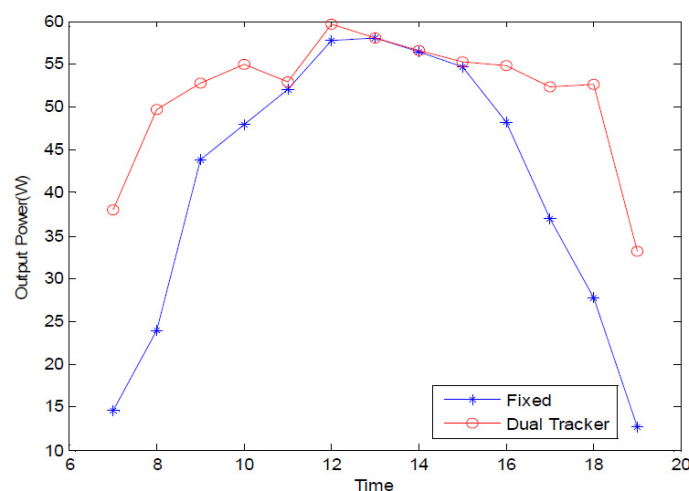


Figura 2.7 - Potência do seguidor de dois eixos e de um sistema com o painel fixo [12]

O acréscimo da energia que se obteve em relação a um sistema fixo foi de aproximadamente 30% para o seguidor de um eixo e 80% para o seguidor de dois eixos [12]. No entanto, tendo em consideração os resultados das medições, recalculou-se esse acréscimo, fora do âmbito deste trabalho, e obteve-se um valor de aproximadamente 45% para o seguidor de dois eixos em relação ao sistema com o painel fixo.

Na Faculdade de Engenharia da Colômbia desenvolveu-se um seguidor de um eixo do tipo horizontal com um servomotor e cinco sensores LDR alinhados na plataforma em arco representada na figura 2.8.

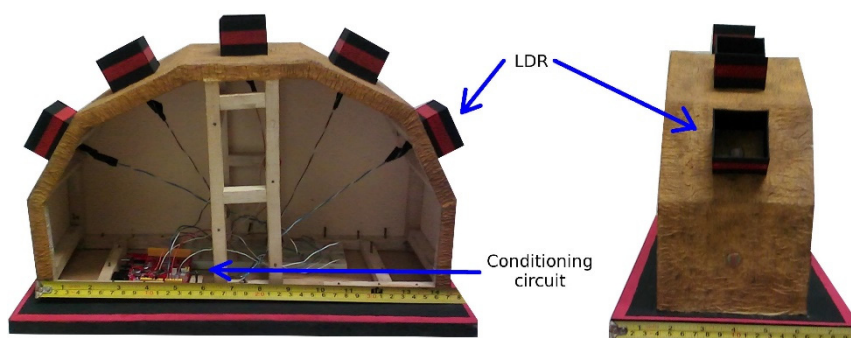


Figura 2.8 - Disposição dos sensores LDR [13]

O sol incide mais num determinado sensor e a correção é efetuada com os sensores vizinhos através da unidade de controlo. Este sistema de sensores é mais eficaz do que os sistemas que utilizam dois sensores. O desempenho foi analisado tendo em consideração dois painéis fotovoltaicos idênticos, um dos painéis foi instalado num sistema fixo com uma inclinação de 15° e o outro foi instalado no seguidor solar. A unidade de controlo foi programada para efetuar medições da tensão e da corrente do painel fotovoltaico em intervalos de dez minutos e as medições decorreram das 08:00 até às 18:00. O acréscimo de desempenho que se obteve com este seguidor foi de 25% em relação ao painel fixo. Os cálculos da potência do painel solar foram efetuados com base na multiplicação da tensão de circuito-aberto com a corrente de curto-circuito [13].

Na Universidade da Malásia desenvolveu um seguidor solar de dois eixos de malha aberta com um módulo GPS, equações astronómicas, dois atuadores lineares DC e três microcontroladores. Neste modelo, o seguidor utiliza equações astronómicas para localizar o posicionamento do sol sem a utilização de sensores de luminosidade. O seguidor utiliza também uma bússola digital que se baseia no norte geográfico (*true north*) para determinar o ângulo de azimuth do sistema. A localização geográfica do seguidor solar é determinada através de um módulo GPS que utiliza satélites para calcular a posição do seguidor. O módulo GPS envia continuamente informação para o microcontrolador, essa informação contém a latitude, longitude, altitude, data e a hora da localização do seguidor. Utilizou-se um microcontrolador principal que interliga dois microcontroladores secundários, responsáveis pelo controlo independente do movimento em

azimute e elevação respetivamente. Este sistema entra num modo de baixo consumo de energia durante a noite. A análise do desempenho do seguidor teve em consideração um sistema com o painel fixo implementado com uma inclinação que pretende gerar a máxima potência no dia do ensaio. A potência máxima é gerada ao meio dia solar e ocorre quando o ângulo de elevação do sol é máximo. As medições foram registadas em intervalos de um minuto a partir das 07:00 até às 20:00. Pela figura 2.9, o acréscimo de energia que se obteve com o seguidor solar em relação ao painel fixo foi de 27% durante um dia sem nuvens [14].

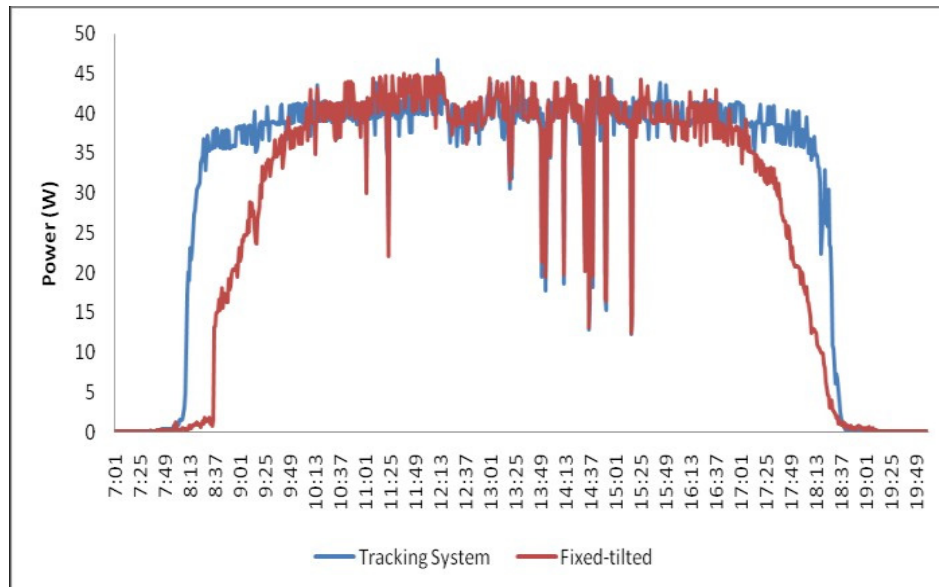


Figura 2.9 - Potência do seguidor solar e de um painel fixo num dia sem nuvens [14]

Também se observou durante um dia com nuvens que o acréscimo de energia que se obteve com o seguidor solar em relação ao painel fixo foi de 19%, como mostra a figura 2.10. [14].

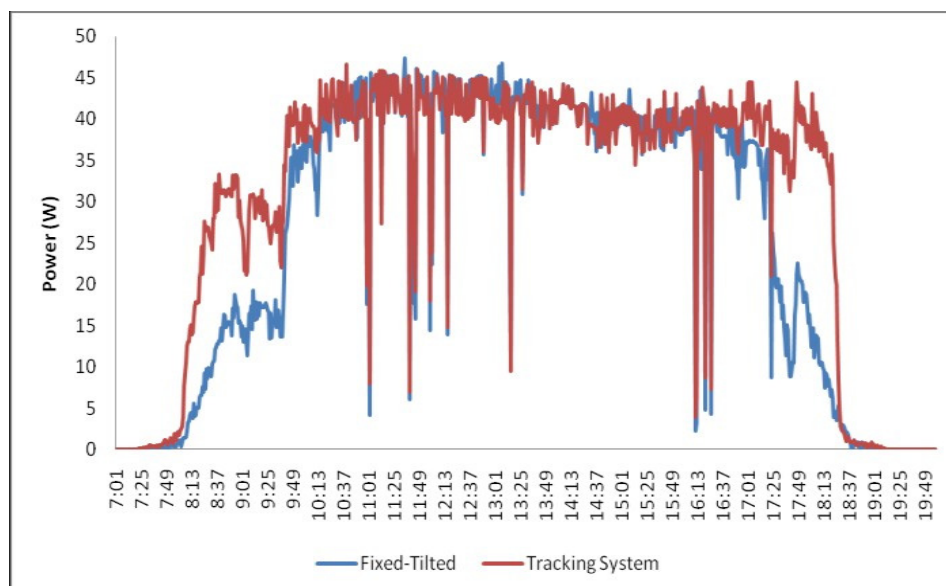


Figura 2.10 - Potência do seguidor solar e de um painel fixo num dia com nuvens [14]

Pela análise dos resultados das dissertações que foram apresentadas neste capítulo, verificou-se que os seguidores solares aumentam o desempenho do painel em relação a um sistema fixo. Esse aumento é de aproximadamente 40% para o seguidor de dois eixos e aproximadamente 30% para o seguidor de um eixo. O desempenho foi também maior no seguidor de dois eixos com sensores de luminosidade [10] em comparação com o seguidor de dois eixos com um algoritmo astronómico [14].

3. Componentes do sistema do seguidor solar

Este capítulo aborda os componentes tipicamente considerados no desenvolvimento de seguidores solares, o modo de operação e os algoritmos de posicionamento solar. Os componentes tipicamente considerados no desenvolvimento de seguidores solares são apresentados a seguir [3]:

- Sistema mecânico: unidade de rotação (*slewing drive*), atuadores lineares, engrenagens helicoidais, parafusos de rosca sem-fim (*worm drive*), engrenagens planetárias;
- Motores elétricos: motores DC ou AC para mover as engrenagens mecânicas;
- Baterias: sistema de baterias para armazenar energia;
- Sensores de luminosidade: sensores de inclinação, LDR e foto-díodos
- Algoritmos de posicionamento solar: algoritmo para calcular o posicionamento do sol em azimute e elevação;
- Unidade de controlo: dispositivo programável (ex.: computador, PLC ou microcontrolador) que coordena os modos de operação, assim como a estratégia de controlo para posicionar o sistema de acordo com o algoritmo de posicionamento solar ou as coordenadas dos sensores;
- Interruptores fim de curso: dispositivos que limitam o movimento mecânico no sentido de prevenir danos no seguidor;
- Sensores atmosféricos ou ambientais: solarímetro, piranómetro, anemómetro, sensor da temperatura ambiente, sensor de humidade, sensor da pressão atmosférica para detetar emergências ou situações climatéricas de risco;
- Coletor solar: módulos fotovoltaicos.

3.1 Sistema mecânico

O movimento de rotação do seguidor solar tem por base um sistema mecânico. Existem sistemas mecânicos distintos, alguns dos quais são feitos por medida, seguidamente é abordado o sistema mecânico designado como a unidade de rotação e os atuadores lineares.

Unidade de Rotação

A unidade de rotação é um dispositivo utilizado frequentemente em seguidores solares para promover o movimento de rotação e pode ser de um eixo ou de duplo eixo. Estes sistemas utilizam parafusos com uma rosca sem-fim (figura 3.2) permitindo que o movimento de rotação seja irreversível, ou seja, bloqueia-se automaticamente. Essa característica torna a unidade de rotação resistente ao vento e outras forças exteriores que atuam no seguidor. A unidade de rotação é construída combinando o rolamento, engrenagem, vedação e outros componentes numa única unidade conforme mostra a figura 3.1. [15]



Figura 3.1 - Unidade de rotação de um eixo [15]

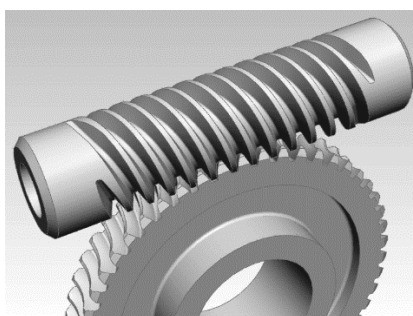


Figura 3.2 – Ilustração do funcionamento da unidade de rotação [16]

Nas unidades de rotação de duplo eixo, combinam-se duas unidades de rotação e dois motores para promover o movimento de rotação horizontal e vertical. [15]

A maioria das unidades de rotação de duplo eixo são produzidas para as centrais elétricas de energia solar, no entanto, também são disponibilizadas unidades de rotação de duplo eixo de menor dimensão para a utilização em habitações como mostra a figura 3.3. [15]

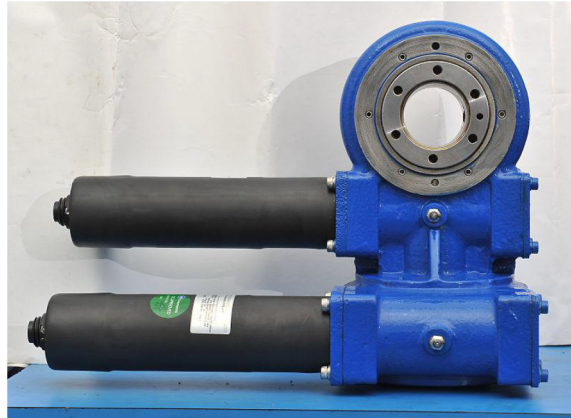


Figura 3.3 - Unidade de rotação de duplo eixo desenvolvida pela Kinematics Manufacturing, Inc [15]

Também existem unidades de rotação que utilizam um sistema com rodas dentadas (figura 3.5) em vez do parafuso com uma rosca sem-fim. Nessa situação, o motor é ligado a uma roda dentada que faz rodar um anel dentado de maior dimensão (figura 3.4). Como este sistema não bloqueia automaticamente, o motor tem de ter outro tipo de mecanismo de bloqueio para suportar as forças exteriores. [15]



Figura 3.4 - Unidade de rotação com rodas dentadas desenvolvido pela IMO [15]



Figura 3.5 - Sistema com rodas dentadas [17]

Atuador Linear

O atuador linear cria movimento em linha reta. Tipicamente é utilizado um parafuso e um motor elétrico que roda esse parafuso para produzir o movimento linear. Os atuadores lineares que utilizam o parafuso Acme bloqueiam automaticamente. Alguns atuadores lineares têm uma resistência variável no interior que muda de valor, dependendo da localização do aperto. Essa informação pode ser utilizada para determinar o posicionamento do seguidor (figura 3.6). [15]

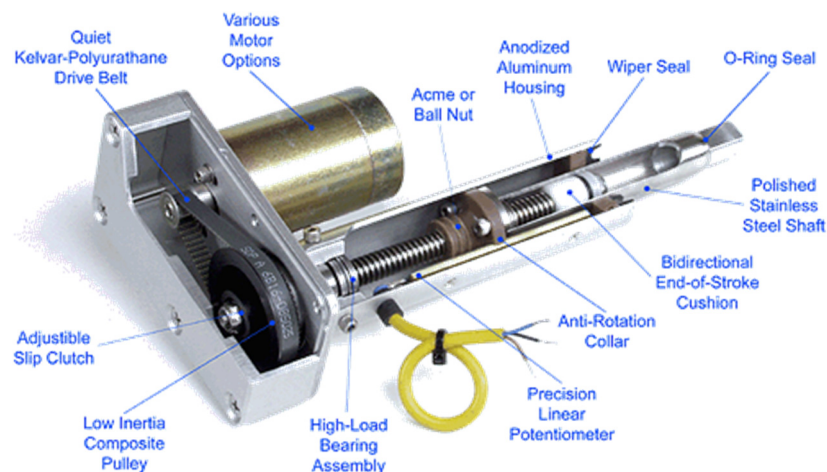


Figura 3.6 - Atuador linear desenvolvido pela Ultramotion [15]

Os atuadores lineares são muito utilizados nos seguidores solares de habitações porque são mais fáceis de controlar e o custo é menor. A inclinação do seguidor pode ser efetuada com atuadores lineares. É frequente a utilização conjunta de atuadores lineares com a unidade de rotação no desenvolvimento de seguidores de dois eixos. [15]

3.2 Motores elétricos

Existe uma variedade de motores elétricos que podem ser utilizados nos seguidores solares. Os motores de indução AC foram mais utilizados nos primeiros seguidores porque se ligam diretamente à rede de energia, em contrapartida, são difíceis de controlar quando a velocidade é baixa. O funcionamento em baixas velocidades é um requisito necessário em grande parte dos seguidores solares [18]. Apesar de os motores passo-a-passo serem capazes de operar em baixas velocidades, a velocidade destes motores pode ser inadequada quando é necessário movimentar rapidamente o seguidor para uma posição segura em situações anómalas ao funcionamento (ex.: tempestades). Os motores BLDC têm uma grande utilidade nos seguidores solares atuais porque não têm escovas diminuindo assim a necessidade de manutenção, são altamente eficientes (tipicamente 85% a 90%) [18].

3.3 Sensores LDR

Os sensores LDR são dispositivos que variam a sua resistência de forma inversamente proporcional à intensidade de luz, sendo assim, podem ser utilizados para determinar o posicionamento solar. Os seguidores de um eixo utilizam tipicamente dois sensores LDR e os de dois eixos utilizar quatro. A distribuição dos sensores é opcional e pode assumir várias configurações sendo essas apresentadas na figura 3.7, como exemplo.

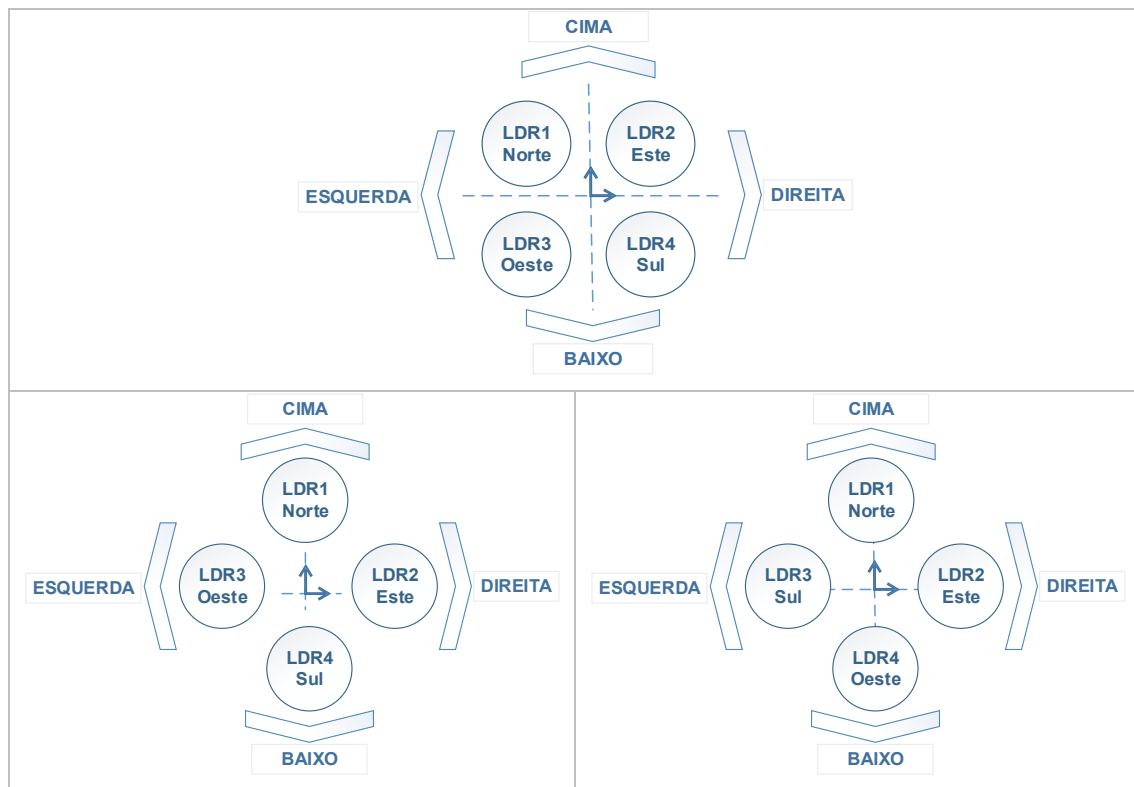


Figura 3.7 – Tipos de agrupamento com sensores LDR num seguidor de dois eixos

Para aumentar a precisão na detecção do posicionamento do sol, colocam-se encapsulamentos ao redor dos sensores LDR. Os sensores formam um ângulo de 45° com a base (figura 3.8) [3].

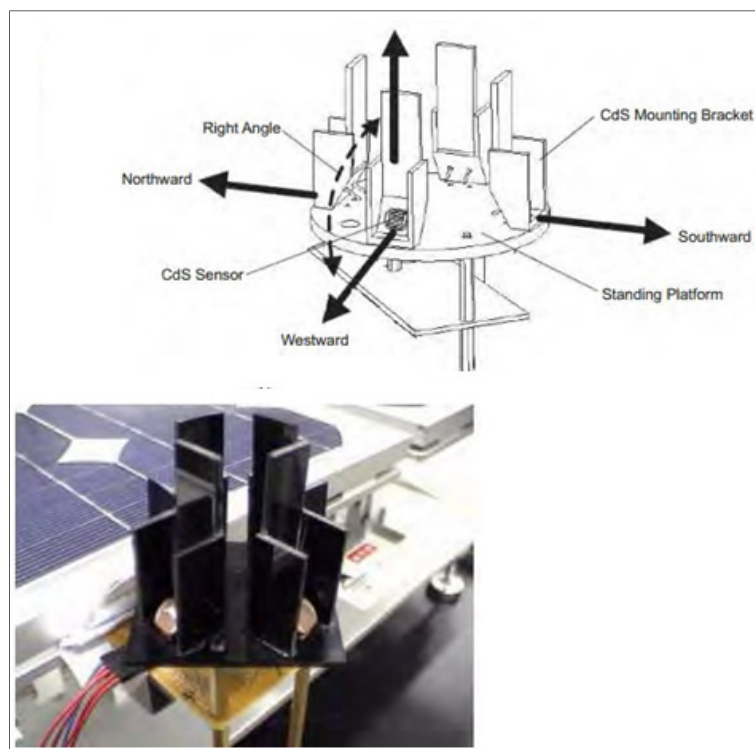


Figura 3.8 - Exemplo da organização dos sensores LDR (CdS) nos seguidores solares [3]

3.4 Unidade de controlo

A unidade de controlo do seguidor solar é um dispositivo programável que pode ser implementado em malha aberta, fechada ou híbrida [3].

Controlo em malha aberta

A tendência atual dos seguidores solares industriais controlados por lógica programável (PLC) é a utilização de controladores em malha aberta [3]. Estes controladores utilizam algoritmos de posicionamento solar (SPA), como o NREL, para direcionar o seguidor solar. O algoritmo NREL SPA é muito utilizado nos seguidores solares sendo um dos mais rigorosos para determinar com exatidão os ângulos solares de elevação, zénite e de azimute [3]. Este algoritmo tem uma margem de erro de $\pm 0,0003^\circ$ referente ao ano de -2000 a 6000, e requer maior capacidade de processamento em comparação com outros algoritmos como o PSA por exemplo [3].

O algoritmo PSA é menos rigoroso ou pode divergir em termos de precisão ao longo do tempo, no entanto, também é utilizado em seguidores solares porque introduz simplicidade e velocidade de processamento [3]. A margem de erro do PSA é $\pm 0,0085^\circ$ para os anos compreendidos entre 1999 a 2015 [19].

Controlo em malha fechada

Nos seguidores com controlo em malha fechada, a realimentação (*feedback*) é promovida por sensores de luminosidade como o LDR ou o Foto-díodo e a informação é analisada pelo controlador no sentido de direcionar o seguidor para o sol. Contudo, os sistemas de seguimento em malha fechada que se baseiam apenas em sensores óticos podem ser afetados pelas nuvens, condições climatéricas ou outros fatores ambientais [3].

Controlo em malha híbrida

O controlo em malha híbrida combina as características da malha fechada e da malha aberta [3]. Assim, os sensores solares são utilizados durante o tempo ensolarado e o algoritmo de posicionamento solar é utilizado quando o sol é coberto pelas nuvens, por exemplo [15].

Modos de operação

Alguns seguidores solares têm implementado um sensor de vento que alerta o controlador para entrar automaticamente no modo de operação em espera (*standby mode*) ou de operação insegura. Nessa situação, a unidade de controlo coloca o painel em posição deitada para minimizar a resistência ao vento [3]. O seguidor também entra no modo de operação em espera quando o mesmo é desligado. Durante a noite, o painel solar é colocado automaticamente em posição deitada e entra num modo de economia de energia evitando assim o consumo de energia por parte dos motores [3].

3.5 Algoritmos de posicionamento solar

Para se utilizar algoritmos de posicionamento solar é necessário conhecer-se a localização geográfica do seguidor solar, a data e a hora. Essas informações são essenciais para se determinar o Vetor Solar e podem ser obtidas com um módulo GPS e um relógio de tempo real (RTC). O Vetor Solar (figura 3.9) descreve o ângulo de elevação e de azimuth solar sendo calculado através da unidade de controle [3].

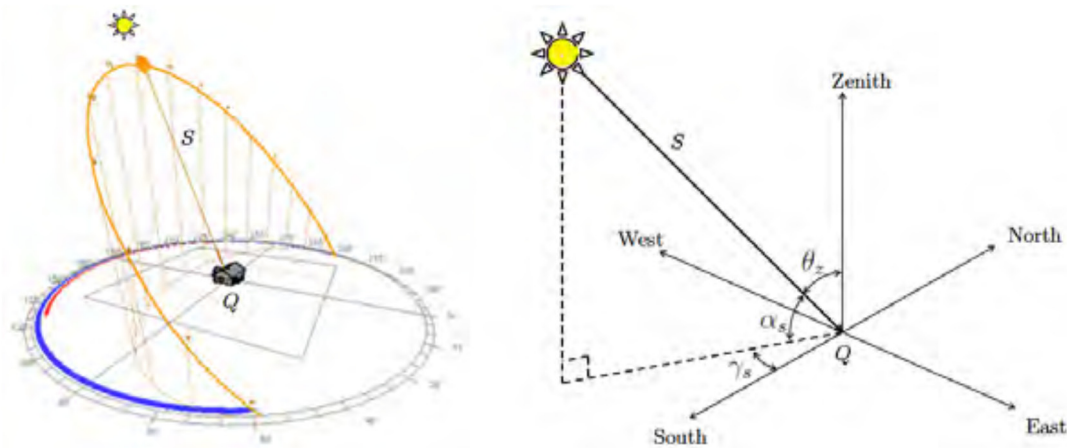


Figura 3.9 - Vetor e ângulos solares com origem na localização Q [3]

O algoritmo de posicionamento solar, quando implementado, devolve as coordenadas do posicionamento do sol em graus e essa informação poderá ser processada para direcionar o seguidor com base em sensores de ângulo e inclinômetros que informam o ângulo atual do seguidor. Assim, o controlador processa o azimuth e a elevação do sol com o algoritmo de posicionamento solar e compara o resultado com a orientação atual do seguidor efetuando depois a rotação necessária para que o painel fique de frente para o sol [3].

4. Implementação do seguidor solar

Este projeto foi implementado com dois motores passo-a-passo para controlar o ângulo de inclinação e de azimute. Os motores que foram utilizados (PD3-109-57) já vêm com um *driver* e um microcontrolador incluído conforme apresentado na figura 4.1.



Figura 4.1 - Motor passo-a-passo PD3-109-57

Estes microcontroladores só têm uma entrada analógica. Sendo assim, para contornar essa limitação, utilizou-se outro microcontrolador, o Arduino Uno. Este microcontrolador adicional recebe a informação analógica proveniente de 4 sensores LDR, processa essa informação e transmite o resultado para o microcontrolador do motor correspondente, seja para virar para a esquerda, direita, cima, baixo ou manter a mesma posição.

A figura 4.2 apresenta de um modo geral a interligação dos dispositivos implementados no seguidor solar que foi desenvolvido.

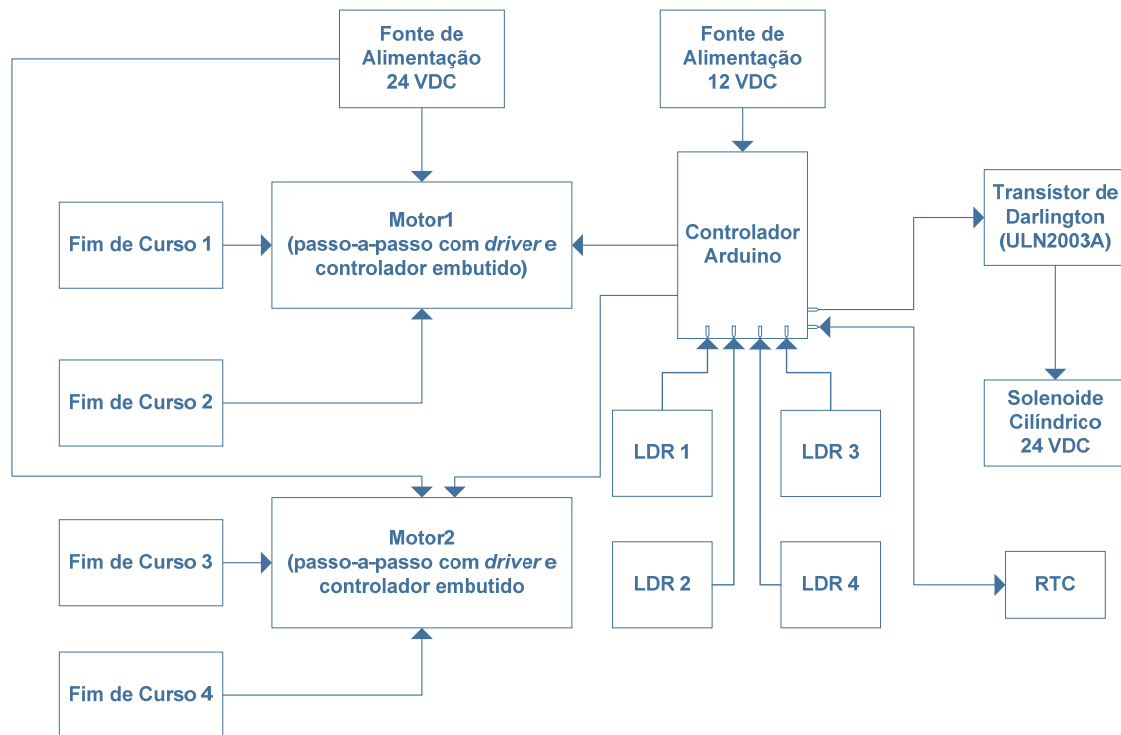


Figura 4.2 - Diagrama de blocos do seguidor solar

Em seguida descreve-se com mais pormenor os dispositivos que foram implementados, como o motor passo-a-passo PD3-109-57 e o controlador Arduino.

4.1 Motor passo-a-passo PD3-109-57

Foram utilizadas duas entradas digitais do microcontrolador de cada motor, terminal GPI_1 e GPI_2 , responsáveis pela ativação da rotação e o controlo da direção de rotação. Essas entradas estão isoladas com acopladores óticos e suportam uma tensão de entrada de 5 V a 24 V (figura 4.3).

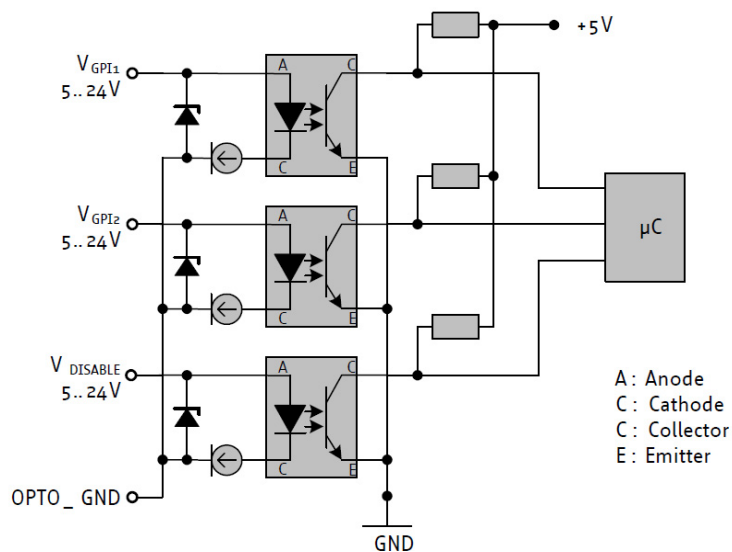


Figura 4.3 – Terminal GPI_1 , GPI_2 e *Disable* do PD3-109-57 [20]

O microcontrolador dos motores passo-a-passo foi programado através da linguagem TMCL [21]. O diagrama de blocos da programação dos motores é apresentado na figura 4.4.

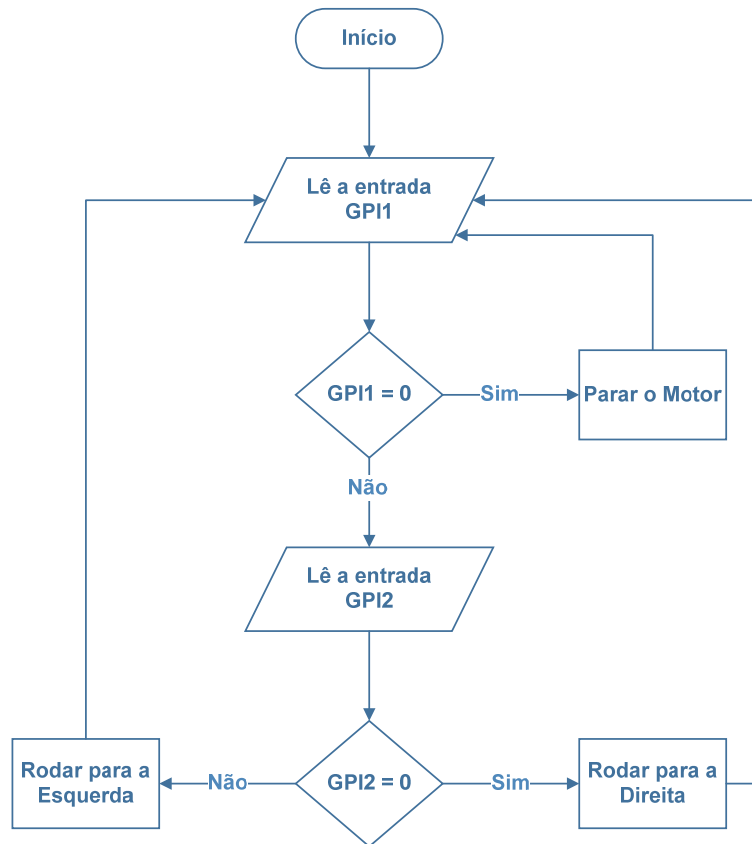


Figura 4.4 - Diagrama de blocos do microcontrolador PD3-109-57

Os Interruptores de Fim de Curso foram ligados em duas entradas dedicadas, *StopR* e *StopL* que ativam interrupções internas no microcontrolador. Estas entradas já possuem internamente resistências de *pull-up* de 10 k Ω (figura 4.5).

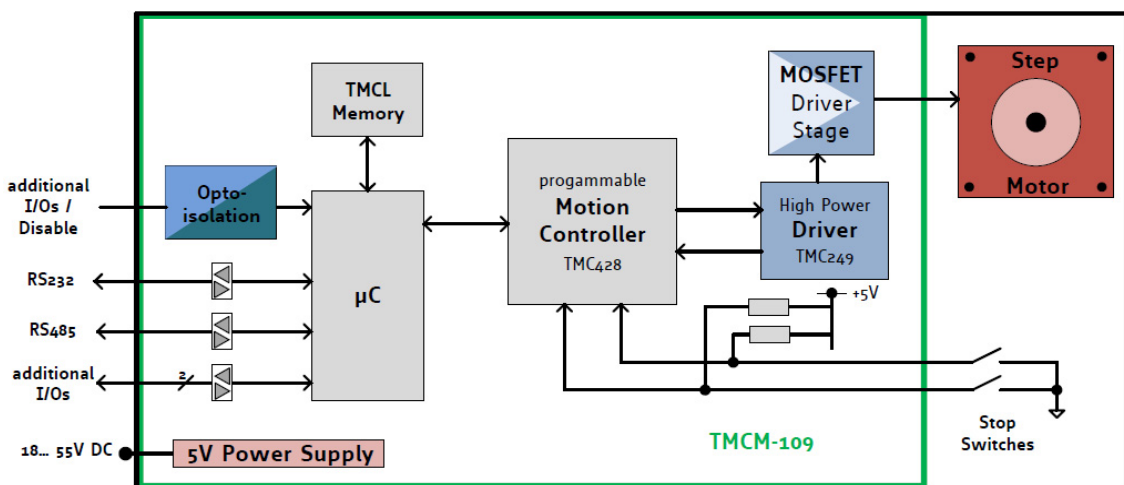


Figura 4.5 – Diagrama de blocos do PD3-109-57 [20]

Para desligar o motor quando este não está a ser utilizado, habilitou-se o terminal *Disable* (*Shutdown*) que ativa uma interrupção para desligar o *driver* e o motor.

4.2 Controlador Arduino Uno

O Arduino Uno é uma placa com 6 terminais analógicos, 14 digitais e um microcontrolador ATmega328P. Este microcontrolador foi programado através da linguagem C++.

Para aumentar a estabilidade na interpretação do sinal dos sensores LDR, procurou-se dar uma tolerância de sensibilidade entre os mesmos. Por exemplo, quando o valor absoluto da subtração do LDR1 (cima) com o LDR4 (baixo) é menor ou igual a um determinado valor configurado no início do programa, o motor permanece imobilizado.

Implementou-se ainda um modo noturno que pode ser ativado quando a intensidade de luz nos quatro sensores LDR atinge um determinado valor configurável no programa. O modo noturno coloca o painel fotovoltaico deitado e desativa os motores. A figura seguinte mostra o algoritmo implementado no Arduino para o seguidor ativo de dois eixos com motores passo-a-passo.

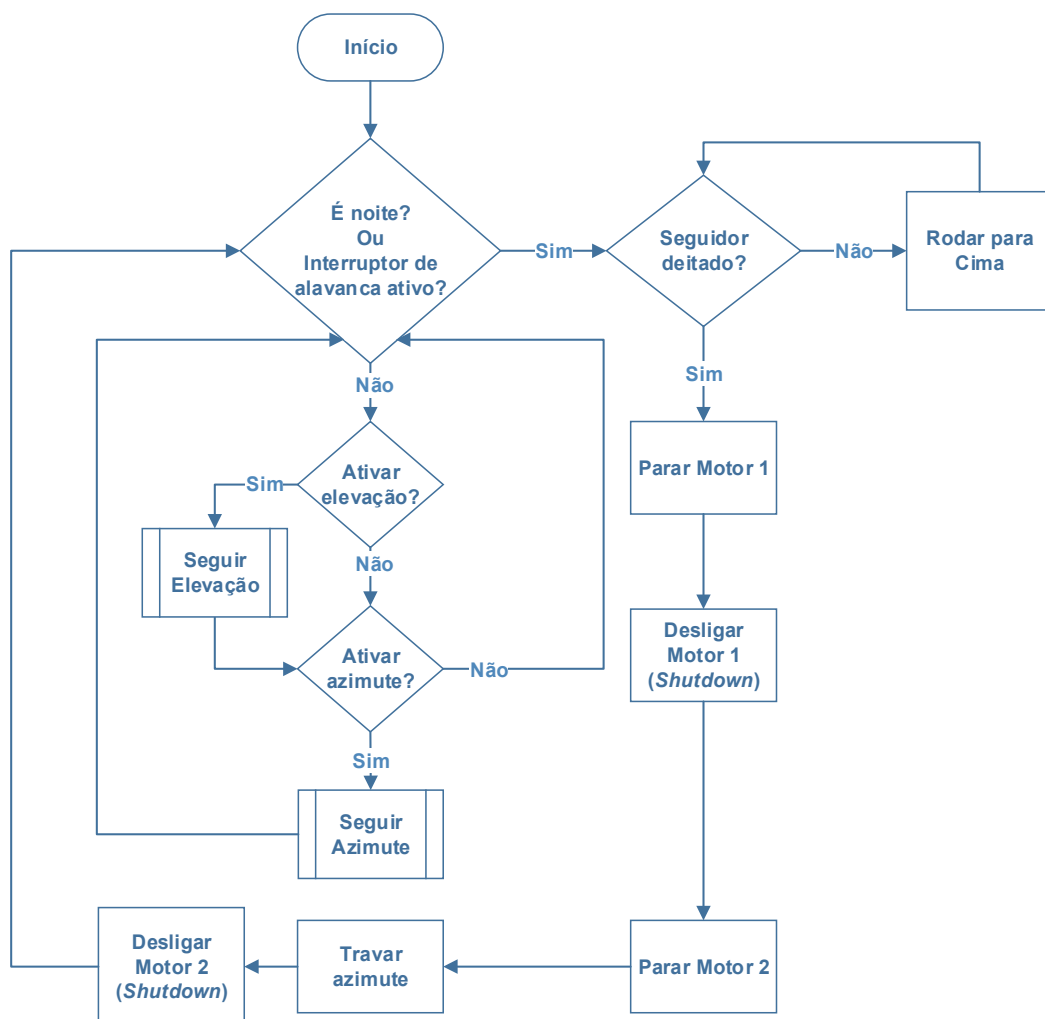


Figura 4.6 - Algoritmo global do Arduino

A figura 4.6 tem duas sub-rotinas, “Seguir Elevação” e “Seguir Azimute”. O diagrama de blocos dessas sub-rotinas é apresentado a seguir nas figuras 4.7 e 4.8.

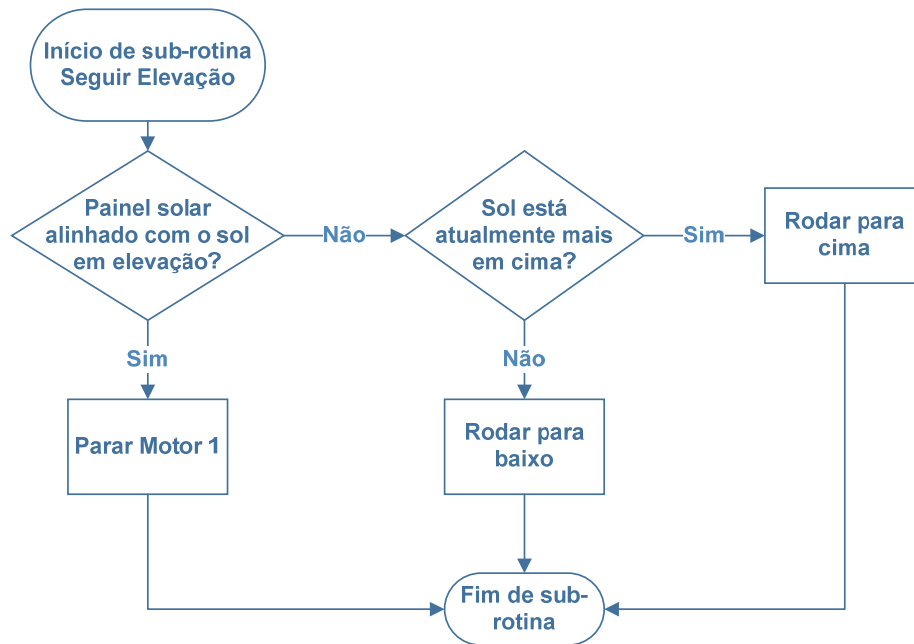


Figura 4.7 - Algoritmo Arduino: sub-rotina Seguir Elevação

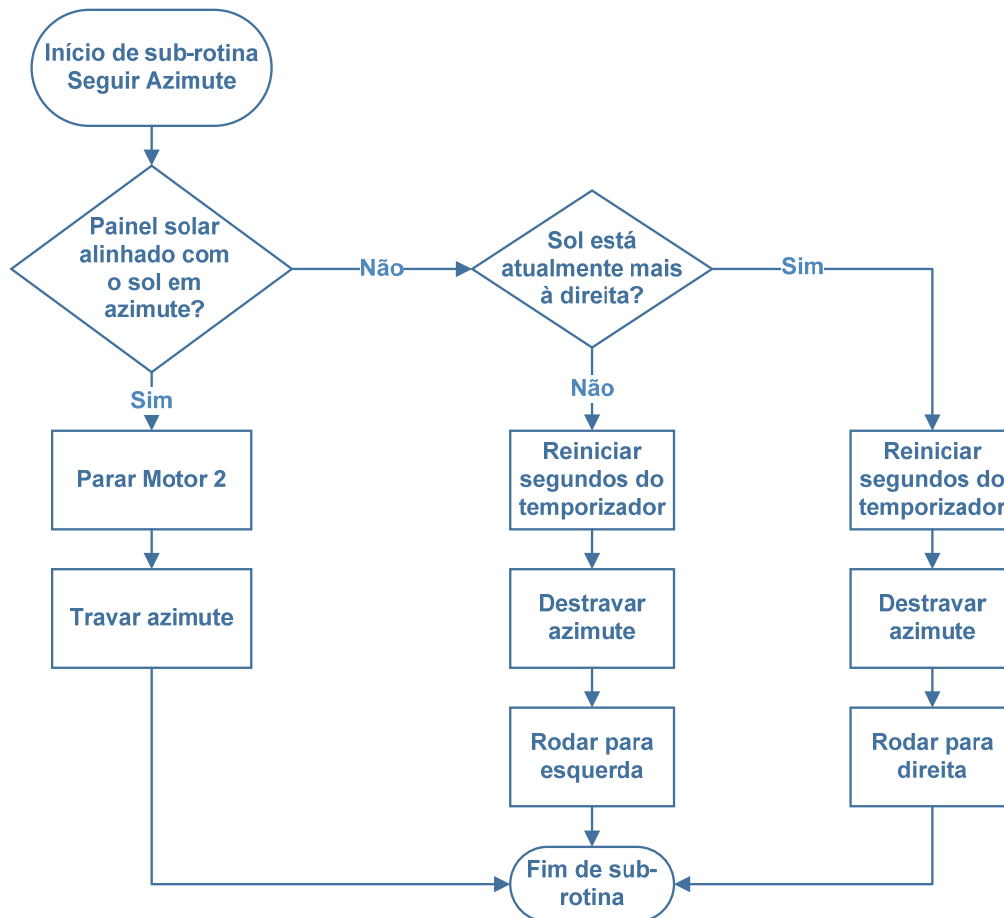


Figura 4.8 - Algoritmo Arduino: sub-rotina Seguir Azimute

Configurou-se um temporizador do Arduino que ativa uma interrupção quando o valor da contagem é igual a um segundo. O motor é desligado automaticamente após 10 segundos de inatividade (figura 4.9).

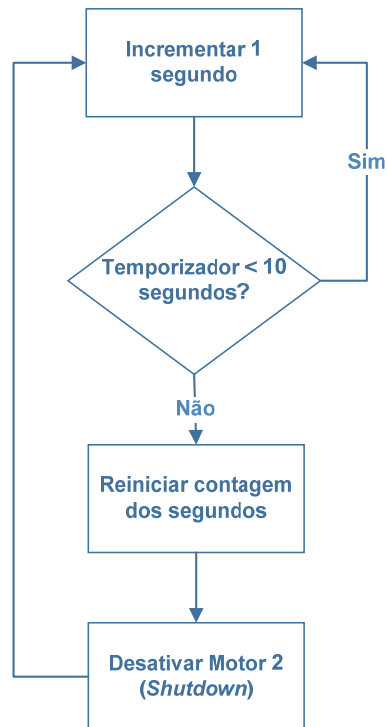


Figura 4.9 – Desativação do motor através do Arduino

4.3 Módulo solar fotovoltaico

O seguidor solar foi projetado para utilizar um módulo Bangkok Solar BS 40, no entanto, quando se testou o módulo num dia ensolarado verificou-se que o mesmo não produzia corrente de curto-circuito estando avariado instalando-se assim um módulo Kyocera KC40. A figura 4.10 mostra o seguidor implementado com o módulo Bangkok Solar.



Figura 4.10 - Seguidor solar desenvolvido com o módulo Bangkok Solar

O teste do módulo foi efetuado direcionado o mesmo para o sol, medindo-se depois a tensão de circuito aberto (V_{OC}) e a corrente de curto-circuito (I_{SC}), os resultados são apresentados na figura 4.1. É espectável que os valores medidos sejam relativamente semelhantes às respectivas características de especificação do módulo [22].

Tabela 4.1 - V_{OC} e I_{SC} ensaiado e especificado para o módulo Kyocera KC40

	Ensaio	Especificação
V_{OC}	19,6 V	21,5 V
I_{SC}	2,25 A	2,48 A

A potência em V_{OC} e em I_{SC} é nula, porque em V_{OC} não existe corrente e em I_{SC} não existe tensão, assim, variou-se a resistência associada ao circuito e optou-se pela implementação da resistência de 25 Ω porque a potência foi maior conforme mostra a tabela 4.2 e a figura 4.11.

Tabela 4.2 -Tensão, corrente e potência com a variação da resistência

Resistência [Ω]	Tensão [V]	Corrente [A]	Potência [W]
∞	19,6	0	0
100	19,04	0,19	3,62
50	18,55	0,37	6,86
25	17,54	0,72	12,63
3	5,69	2,16	12,29
0	0	2,25	0

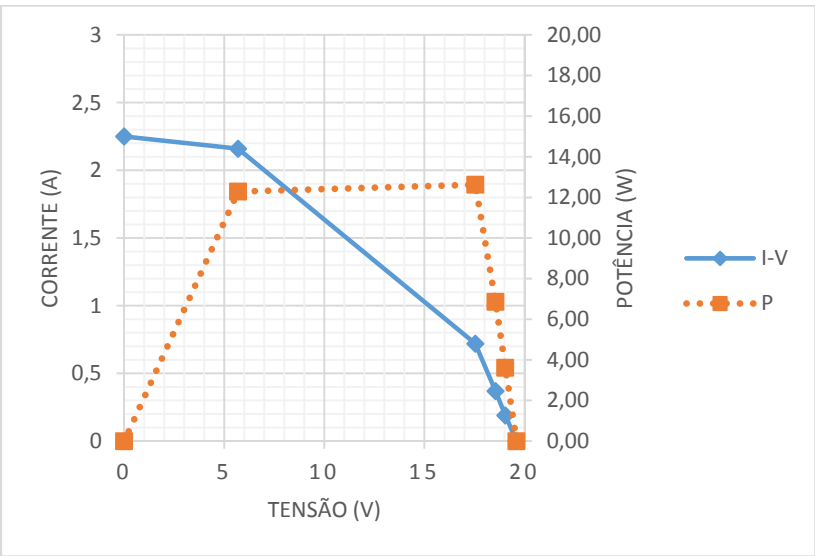


Figura 4.11 - Curva I-V e Potência, referente à tabela 4.2

Tendo em consideração que o ponto de potência máxima (M_{pp}) do módulo fotovoltaico varia em função das condições climáticas, da temperatura e da quantidade de luz que chega ao módulo [23], pode ser mais adequado efetuar-se uma estimativa do M_{pp} tendo em consideração que a tensão na potência máxima (V_{mp}) tem um valor na ordem dos 80% a 90% da tensão de circuito aberto (V_{oc}) e a corrente na potência máxima (I_{mp}) tem um valor na ordem dos 85% a 95% da corrente de curto-circuito (I_{sc}), onde $P_{max} = V_{mp} \times I_{mp}$ (figura 4.12) [24].

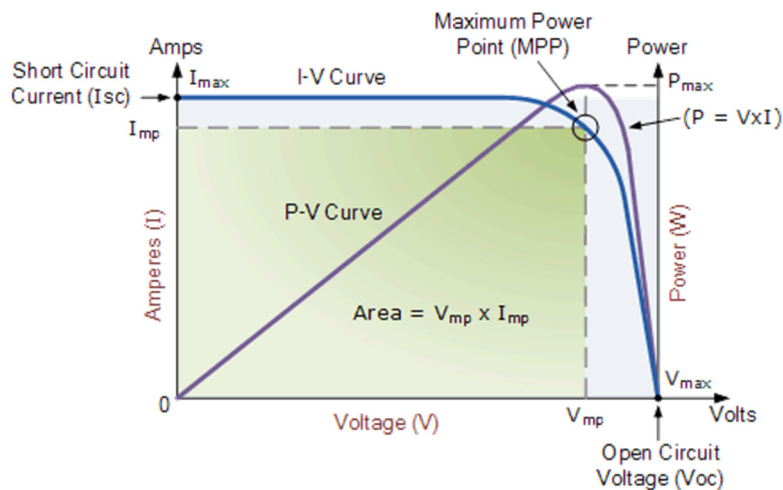


Figura 4.12 - Curva I-V de uma célula fotovoltaica de silício

4.4 Sensores de luminosidade

Os sensores foram encapsulados com um pequeno invólucro para minimizar o efeito da radiação solar difusa, ou seja, a propagação de luz que é dispersa no meio envolvente. A inclinação de 45° foi testada em laboratório e nas condições de teste não se verificou uma melhoria significativa em termos de precisão aliado à dificuldade de calibração. Assim, optou-se por se instalar os sensores sem inclinação. Para aumentar a precisão, colocou-se um segundo encapsulamento semiaberto em torno de cada sensor para aumentar o sombreamento quando o sol se desloca de posição. Os sensores foram organizados em conformidade com a figura 4.13.

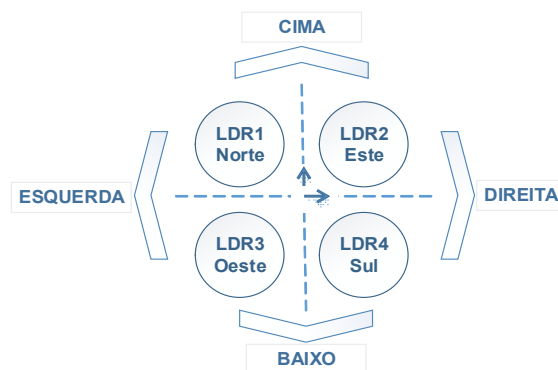


Figura 4.13 - Distribuição dos sensores LDR

Implementaram-se divisores de tensão com os sensores LDR (figura 4.14), onde as saídas dos divisores de tensão são ligadas às entradas analógicas do controlador. Se forem detetadas diferenças, os motores são acionados num determinado sentido, se não houver diferença entre os sinais à saída dos divisores de tensão, então o painel fotovoltaico está direcionado para o sol [3].

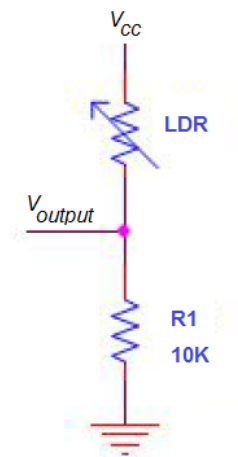


Figura 4.14 - Divisor de Tensão com LDR

A saída do divisor de tensão é dada pela relação:

$$V_{output} = V_{cc} \times \left(\frac{R_1}{LDR + R_1} \right)$$

Para se ter uma percepção da variação de tensão à saída do divisor, mediu-se com um multímetro a resistência do sensor LDR em cenários de luminosidade diferentes. A tabela 4.3 mostra os valores obtidos para um R_1 de 10 k Ω , um V_{cc} de 5 V.

Tabela 4.3 - Teste da resistência do LDR em cenários diferentes de luminosidade

Cenário	Resistência do LDR	V_{output}
Pouca luz (LDR tapado)	320 k Ω	0,15 V
Luz normal do ambiente	5,6 k Ω	3,2 V
Muita luz (enfoque de uma lanterna em frente ao LDR)	107 Ω	4,9 V

4.5 Algoritmos de posicionamento solar

Conforme mencionado no capítulo 3, o algoritmo NREL SPA tem uma precisão maior do que o PSA. Tentou-se implementar o NREL e verificou-se que o controlador utilizado (Arduino Uno) não tem capacidade para processar esse algoritmo, assim implementou-se o PSA. A data e a hora foram estabelecidas por um módulo RTC e as coordenadas geográficas (latitude, longitude) foram programadas manualmente no Arduino em vez da utilização de um módulo GPS.

Para se efetuar o posicionamento do seguidor, optou-se pela utilização de um temporizador interno do controlador. Por falta de portas disponíveis no controlador, não se utilizou o sensor de ângulo nem o inclinômetro conforme mencionado no capítulo anterior. Numa primeira fase, efetuou-se uma calibração através da medição dos graus percorridos pelo seguidor durante um segundo, tanto para o azimute como para a elevação, depois fez-se uma estimativa com essa informação em relação ao tempo que seria necessário o motor girar para atingir uma determinada posição especificada pelo algoritmo.

Nesta configuração, o seguidor deve de ser inicializado com o interruptor de alavanca desligado ou com o painel solar deitado. Quando o interruptor de alavanca é ligado, o controlador calcula a elevação do sol em relação ao posicionamento atual do seguidor e direciona o painel solar nesse sentido. Simultaneamente, o seguidor roda para a esquerda até alcançar o interruptor fim-de-curso, sendo depois calculado o posicionamento do sol em azimute e o painel é direcionado nesse sentido.

4.6 Sistema de bloqueio automático

Como o sistema mecânico implementado utiliza rodas dentadas, foi necessário desenvolver um mecanismo de bloqueio automático para suportar as forças exteriores e para desativar os motores em segurança no sentido de economizar energia. Neste caso, procurou-se utilizar dois solenoides cilíndricos para bloquear o movimento em azimute e elevação respetivamente. O solenoide quando magnetizado liberta a roda dentada para o motor fazer girar o seguidor (figura 4.14).



Figura 4.15 - Solenoide cilíndrico 24 V_{DC} para o bloqueio do azimute

Após a implementação, verificou-se que o solenoide responsável pelo bloqueio da inclinação não desacoplava em algumas situações, por causa do atrito gerado com a roda dentada do seguidor. Esta situação acontecia com mais frequência quando o painel solar se encontrava em desequilíbrio entrando em conflito com o motor que não conseguia rodar quando havia necessidade. Sendo assim, removeu-se o solenoide e desativou-se o temporizador que desabilita o motor associado à inclinação. O solenoide para o bloqueio do azimute foi implementado, assim como o integrado UL2003 contendo transístores de Darlington para fazer a interligação entre a saída de 5 V_{DC} do controlador e os 24 V_{DC} do solenoide. Estes transístores já incluem díodos de roda livre no sentido de evitar picos de tensão quando o solenoide é desligado.

Construiu-se uma placa para acomodar o UL2003 e os divisores de tensão com resistências de 10 k Ω que interligam os sensores LDR através de um cabo RJ45 removível.

4.7 Caixa do seguidor solar

A figura 4.15 mostra os componentes que foram instalados na caixa do seguidor solar, nomeadamente, a fonte de alimentação de 24 V_{DC}, a placa de interligação que foi desenvolvida, o controlador Arduino Uno e o módulo RTC.

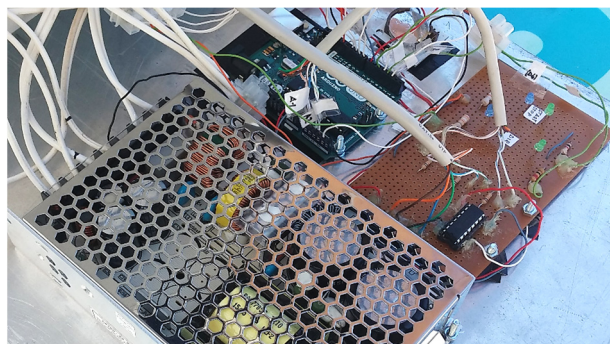


Figura 4.16 - Constituintes internos da caixa do seguidor solar

Colocou-se um interruptor de alavanca ligado ao controlador para desligar o seguidor solar em segurança, o painel primeiro fica deitado e depois os motores são desligados.

5. Resultados e discussão

Os testes decorreram no dia 16 de Julho de 2016 na latitude 38,660073 e longitude -9,204856. Nesse dia, o céu não tinha nuvens e o sol começou a incidir diretamente no painel solar pouco depois das 10 horas da manhã até às 17 horas aproximadamente.

O ângulo ótimo de inclinação anual do painel solar fixo foi determinado com a aplicação PVGIS disponível na internet [25] e obteve-se um ângulo de 34°. A análise do desempenho teve em consideração as medições da tensão e da corrente do painel fotovoltaico associado a uma resistência de 25 Ω , calculando-se depois a potência em intervalos de uma hora em relação aos seguintes cenários da tabela 5.1:

Tabela 5.1 - Condições de teste do seguidor solar com sensores e algoritmos SPA

Ensaio	Condição inicial de teste
Dois eixos	<ul style="list-style-type: none">Sem intervenção manual
Seguimento apenas da elevação do sol	<ul style="list-style-type: none">Painel solar com azimuth fixo: virado para sul
Seguimento apenas do azimuth do sol	<ul style="list-style-type: none">Painel solar com inclinação fixa: 34°
Fixo	<ul style="list-style-type: none">Painel solar com azimuth fixo: virado para sulPainel solar com inclinação fixa: 34°

Os resultados que se obteve para a potência de saída do módulo fotovoltaico nos diferentes cenários de teste são representados graficamente nas figuras 5.1 e 5.2.

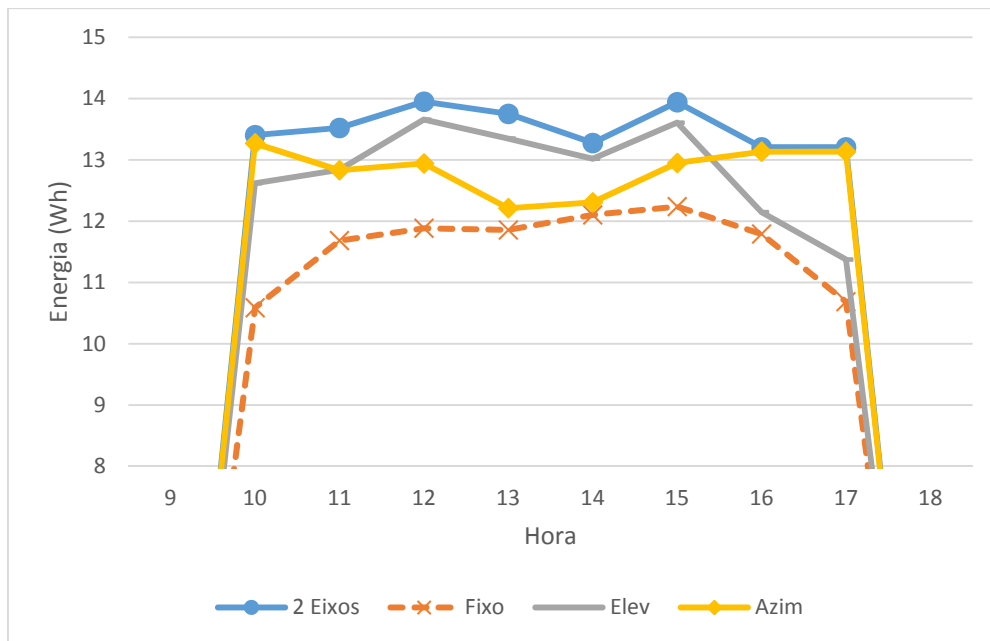


Figura 5.1 – Conversão energética do módulo no seguidor solar com sensores LDR

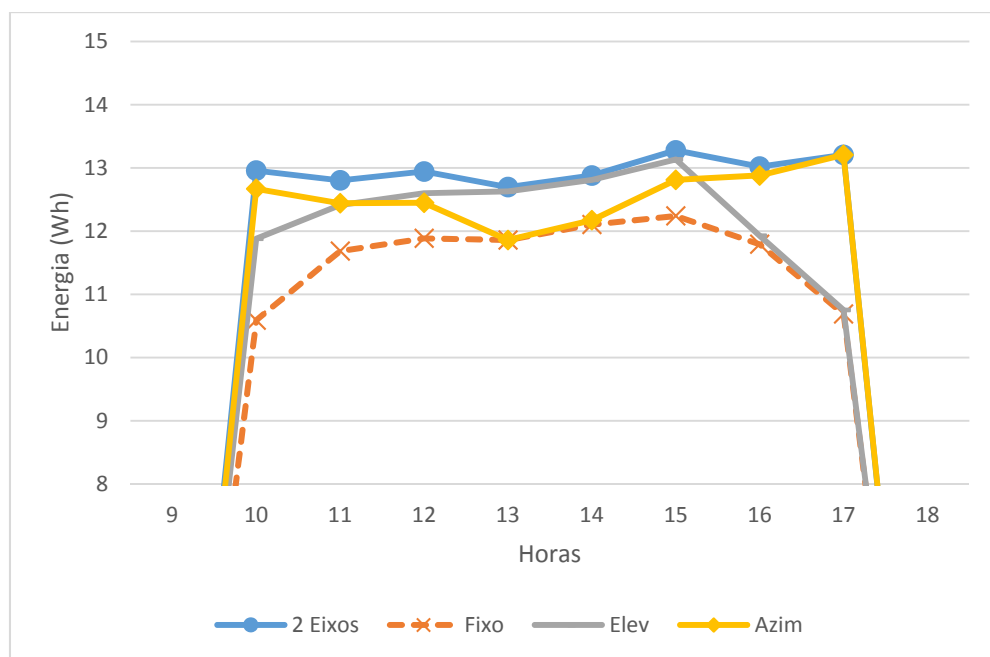


Figura 5.2 – Conversão energética do módulo no seguidor solar com o algoritmo PSA

No geral, a energia que se obteve com o seguidor a funcionar apenas com sensores LDR foi maior do que com o algoritmo PSA. Esta situação poderá estar relacionada com a calibração do seguidor em relação ao tempo de rotação dos motores para alcançar um determinado ângulo e também com a calibração dos sensores LDR. Em condições laboratoriais, a margem de erro do seguidor a funcionar com o algoritmo PSA foi de aproximadamente 2° para a elevação, e 4° para o azimute.

Pela comparação dos resultados entre os seguidores de um eixo, verificou-se no caso do seguidor em elevação que o desempenho é maior a meio da tarde e menor no início e final do dia; verificou-se o oposto em relação ao seguidor em azimute, o desempenho foi maior ao início e ao final da tarde, sendo menor a meio da tarde. A análise do desempenho do seguidor solar em relação ao painel fixo teve em consideração o acréscimo percentual da potência do painel quando instalado no seguidor.

$$\text{Acréscimo de potência [\%]} = \frac{\text{Potência do seguidor solar} - \text{Potência do painel fixo}}{\text{Potência do painel fixo}} \times 100$$

A tabela 5.2 mostra o acréscimo de potência do seguidor solar ao longo do dia em comparação com um sistema fixo.

Tabela 5.2 - Comparação do desempenho do painel do seguidor solar em relação a um sistema fixo

Hora	Seguidor solar com sensores LDR			Seguidor solar com algoritmo PSA		
	2 Eixos	Elevação	Azimute	2 Eixos	Elevação	Azimute
10:00	26,6%	19,2%	25,3%	22,3%	12,2%	19,6%
11:00	15,7%	9,9%	9,8%	9,6%	6,2%	6,5%
12:00	17,4%	14,9%	8,9%	8,9%	6,0%	4,7%
13:00	16,0%	12,6%	3,0%	7,1%	6,5%	0,0%
14:00	9,7%	7,6%	1,7%	6,4%	5,8%	0,6%
15:00	13,9%	11,2%	5,8%	8,5%	7,3%	4,7%
16:00	12,0%	3,0%	11,4%	10,4%	1,1%	9,2%
17:00	23,6%	6,4%	22,9%	23,6%	0,6%	23,6%
Média Total	16,8%	10,6%	11,1%	12,1%	5,7%	8,6%

Pelo que foi mencionado no capítulo 2, o desempenho anual do seguidor solar em comparação com um sistema fixo aumenta até 40% com dois eixos e 30% com um eixo. Experimentalmente, o máximo que se conseguiu obter foi aproximadamente 20% com dois eixos e 10% com um eixo. Considerando que na data e localização de teste o sol nasce às 6:38 da manhã e põe-se às 20:50 [26], o desempenho do seguidor poderia ser maior porque só foi possível efetuar medições das 10:00 às 17:00 por causa de obstáculos como árvores que bloqueavam a luz do sol. Também é possível verificar que o desempenho é tendencialmente maior durante a manhã e ao final da tarde, como exemplifica a figura 5.3.

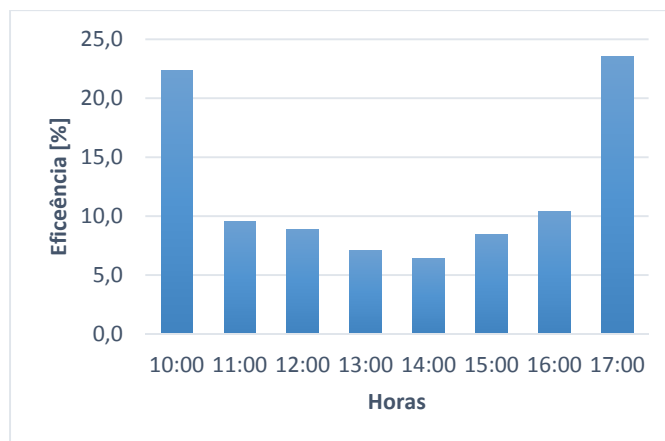


Figura 5.3 – Acréscimo de potência do seguidor solar PSA de 2 eixos em relação a um sistema fixo

Análise dos pontos de cruzamento do gráfico de desempenho PSA

Pelo gráfico da figura 5.2, a potência do painel instalado no seguidor do azimute é sensivelmente igual à do sistema fixo pelas 13 horas e a potência do painel instalado no seguidor da elevação é igual à do sistema fixo pelas 17 horas. O gráfico do seguidor do azimute cruza o gráfico do seguidor da elevação sensivelmente às 11 horas e às 15 horas. Para analisar estas situações, recorreu-se a uma aplicação disponível na internet [27] que permite simular os resultados do algoritmo PSA na data e localização geográfica do estudo, sendo os resultados apresentados na tabela 5.3.

Tabela 5.3 - Simulação dos resultados do algoritmo PSA

Hora	Elevação	Azimute
07:00	16,31°	75,52°
08:00	27,82°	84,09°
09:00	39,51°	93,32°
10:00	51,07°	104,58°
11:00	61,88°	120,83°
12:00	70,29°	149,03°
13:00	72,19°	193,10°
14:00	65,96°	229,03°
15:00	55,86°	249,27°
16:00	44,51°	262,14°
17:00	32,82°	272,01°
18:00	21,19°	280,74°
19:00	9,89°	289,29°
20:00	-0,83°	298,32°
21:00	-10,62°	308,36°

Estes resultados são idênticos aos que foram gerados pelo controlador Arduino no dia dos testes experimentais. A tabela 5.3 é representada graficamente nas figuras 5.4 e 5.5.

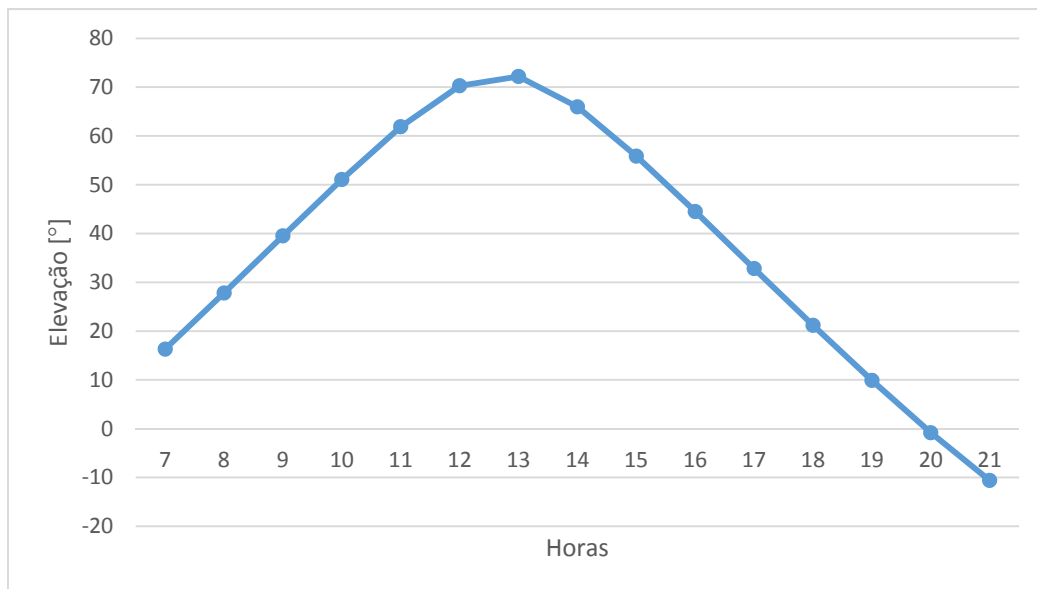


Figura 5.4 – Ângulo de elevação solar obtido com o algoritmo PSA

No cenário de teste, o painel quando fixo encontra-se virado para sul com uma inclinação de 34° aproximadamente. Pelo gráfico da figura 5.4, o ângulo de elevação solar máximo ocorre pelas 13 horas. O ângulo de elevação solar é igual a 34° pelas 8:30 e as 17 horas, este ângulo corresponde à inclinação do painel quando fixo, estando em conformidade com os resultados obtidos na figura 5.2.

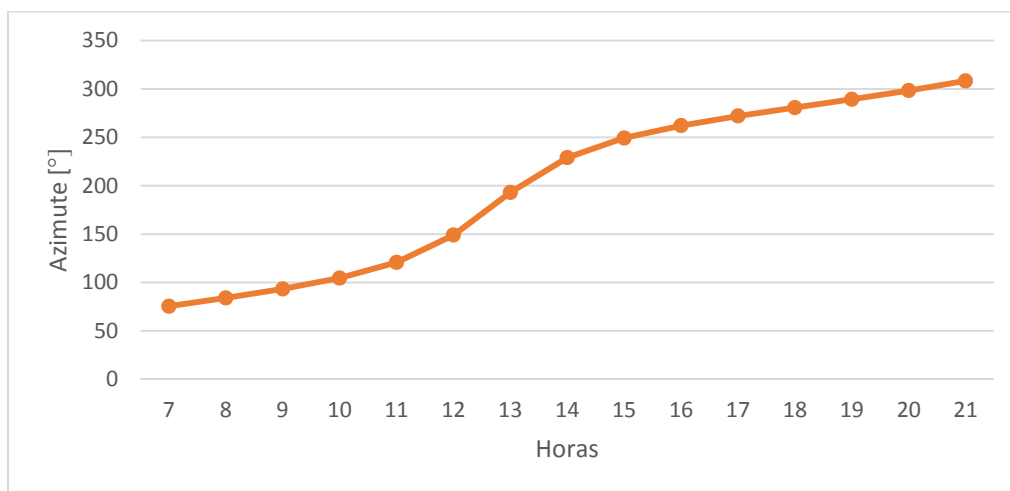


Figura 5.5 – Ângulo de azimute solar obtido com o algoritmo PSA

Podemos verificar no gráfico da figura 5.5 que o sol se encontra virado para o sul (180° do norte) aproximadamente às 13 horas, ou seja, fica de frente para o seguidor. Nessa situação, a potência

do painel do seguidor é semelhante ao sistema com o painel fixo quando se ativa apenas o seguimento em azimuth porque a inclinação do painel é a mesma em ambas as situações.

O gráfico do azimuth, representado na figura 5.5, foi modelado para apresentar apenas valores absolutos no sentido de se perceber em que altura a variação da rotação em azimuth é igual à variação da rotação em elevação. Como os resultados teóricos do azimuth são em relação ao norte (0° do norte) e o sistema está virado por defeito para sul (180° do norte), subtraiu-se 180° aos resultados do azimuth.

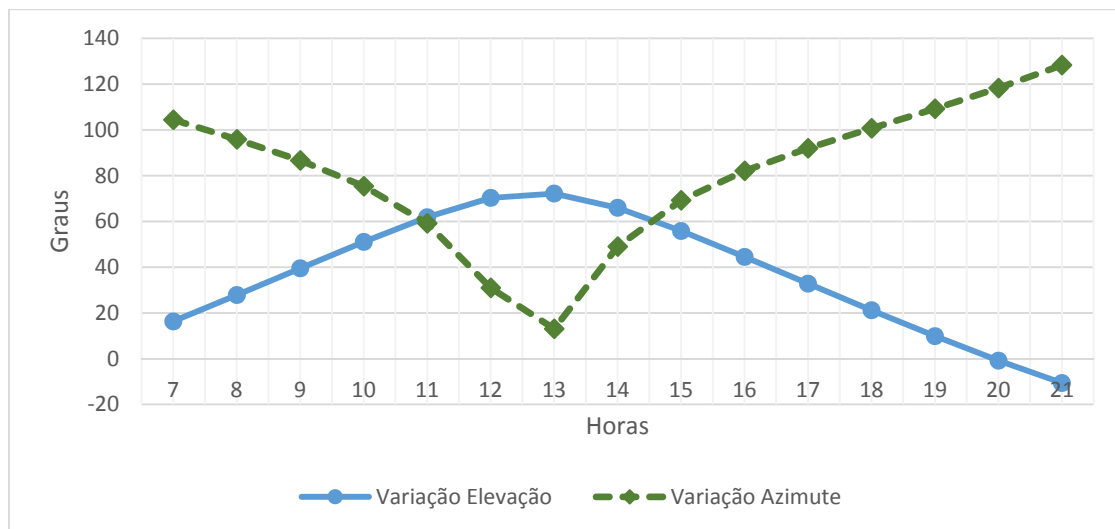


Figura 5.6 - Variação do azimuth e da elevação com o seguidor virado inicialmente para sul

Pela figura 5.6, o movimento de rotação do azimuth e da elevação é semelhante pelas 11 horas e as 14:30. Em relação aos resultados da figura 5.2, as potências do seguidor em azimuth e do seguidor em elevação são semelhantes pelas 11 horas e as 15 horas. Assim, de um modo geral, os resultados obtidos estão em conformidade com os dados teóricos do algoritmo PSA.

6. Conclusão

A procura crescente por módulos fotovoltaicos tem aumentado a procura por seguidores solares. Tendo em consideração um painel fixo, os fabricantes de seguidores solares anunciam um aumento de desempenho de 40% se o seguidor for de dois eixos, ou 30% se for de um eixo. O aumento do desempenho traduz-se também numa otimização do espaço disponível de instalação do sistema fotovoltaico porque é necessário menos espaço para gerar a mesma quantidade de energia em comparação com a instalação de um painel solar fixo. Considerando a localização geográfica, os seguidores de um eixo do tipo horizontal são mais adequados em baixas latitudes como nos trópicos e perto do equador, e os do tipo polar são mais adequados em latitudes altas, afastados do equador e dos trópicos. Apesar da adaptabilidade dos seguidores solares, os painéis solares fixos conseguem suportar condições ambientais mais adversas e necessitam de menos manutenção. Ainda assim, o custo dos seguidores solares de dois eixos é maior aleado ao acréscimo da complexidade do sistema. Nesse contexto, desenvolveu-se e implementou-se um seguidor solar de dois eixos com motores passo-a-passo, sensores de luminosidade e algoritmos de posicionamento solar para se analisar o desempenho do seguidor de dois eixos e de um eixo com a variação apenas do azimute e da inclinação, no sentido de se comparar os resultados com o desempenho de um painel fixo. Este trabalho teve também o objetivo de analisar o desempenho do seguidor quando o sistema de seguimento solar é baseado em sensores de luminosidade e quando o sistema de seguimento solar é horário (sem sensores de luminosidade). Como estratégia, utilizou-se apenas um módulo fotovoltaico instalado no painel solar do seguidor e a análise do desempenho teve por base as medições da tensão, corrente e o cálculo da potência de saída do módulo associado a uma resistência de potência. O ganho percentual do desempenho, em comparação com um sistema fixo, é tendencialmente maior durante a manhã e ao final da tarde. No geral, para a localização geográfica do estudo, o módulo fotovoltaico do seguidor de dois eixos tem uma potência de saída que se situa próxima do máximo para o painel, ao longo do dia; no caso do seguidor de um eixo

com seguimento em azimute, a potência é maior ao início e ao final da tarde; o seguimento em elevação permitiu obter maior potência a meio da tarde. O seguidor com sensores de luminosidade foi mais eficiente do que o seguidor horário, no entanto, dadas as condições de teste e a dificuldade encontrada na calibração dos dispositivos, os resultados podem não ser conclusivos a esse respeito. Assim, concluiu-se que de uma forma geral, a potência do painel fotovoltaico é maior com seguidores solares e, para a localização do estudo, a potência do painel fotovoltaico é maior com a utilização de um seguidor solar de dois eixos, menor com seguimento apenas em azimute e menor ainda com seguimento apenas em elevação. Em última análise, o cenário que apresenta menor desempenho corresponde à instalação do painel solar num sistema fixo.

Desenvolvimentos futuros

No que respeita a trabalhos futuros, como os testes decorreram num dia sem nuvens, pode ser necessário efetuar-se testes adicionais com outras condições climáticas e repetir a medição do desempenho em dias diferentes no sentido de se obter uma estimativa anual mais rigorosa. Este trabalho teve em consideração a análise do desempenho com a unidade de controlo em malha fechada e aberta, poderia também ser considerado futuramente o desempenho em malha híbrida.

Bibliografia

- [1] A. A. H. Ali, Artist, *2-Axis Solar Tracker Model*. [Thesis]. University of Khartoum, 2013.
- [2] E. Wesoff, "Greentech Media," 22 Fevereiro 2016. [Online]. Available: <http://www.greentechmedia.com/articles/read/SunPower-Breaks-Solar-Panel-Efficiency-Record-Again>. [Acedido em 18 Julho 2016].
- [3] G. Prinsloo e R. T. Dobson, em *Solar Tracking*, 2014, pp. 7-14; 22; 34-35; 65-107; 117-123; 127-130; 143; 150-152.
- [4] J. S. Young , "UMass Amherst," Dezembro 2010. [Online]. Available: <http://www.umass.edu/sunwheel/pages/moonteaching.html>. [Acedido em 11 Março 2016].
- [5] Sandia Corporation, "PVPMC," U.S. Department of Energy's National, 2014. [Online]. Available: <https://pvpmc.sandia.gov/modeling-steps/1-weather-design-inputs/sun-position/>. [Acedido em 10 Março 2016].
- [6] M. R. Store, "Market Research Store," 6 Janeiro 2016. [Online]. Available: <http://www.marketresearchstore.com/news/global-solar-tracker-market-146>. [Acedido em 21 Julho 2016].
- [7] S. C. Staff, "Solar Choice," 21 Janeiro 2010. [Online]. Available: <http://www.solarchoice.net.au/blog/solar-trackers/>. [Acedido em 15 Abril 2016].

- [8] V. N. Keskar, "Electricity Generation Using Solar Power," *International Journal of Engineering Research & Technology (IJERT)*, vol. II, pp. 3-5, 2013.
- [9] S. Bushong, "Solar Power World," 9 Maio 2016. [Online]. Available: <http://www.solarpowerworldonline.com/2016/05/advantages-disadvantages-solar-tracker-system/>. [Acedido em 21 Julho 2016].
- [10] E. M. Assaf, "Design and Implementation of a Two Axis Solar Tracking System Using PLC Techniques by an Inexpensive Method," *International Journal of Academic Scientific Research*, vol. 2, nº 3, pp. 54-65, 2014.
- [11] S. Rahman, R. A. Ferdaus, M. A. Mannan e M. A. Mohammed, "Design & Implementation of a Dual Axis Solar Tracking System," *American Academic & Scholarly Research Journal*, vol. 5, pp. 52-53, 2013.
- [12] R. Dhanabal, Bharathi.V., R. R., A. Ponni, S. Deepthi e P. Mageshkannan, "Comparison of Efficiencies of Solar Tracker systems with static panel Single-Axis Tracking System and Dual-Axis Tracking System with Fixed Mount," *International Journal of Engineering and Technology (IJET)*, vol. 5, nº 2, p. 1927, 2013.
- [13] Y. Garcia, O. Diaz e C. Agudelo, Artists, *Performance of a solar PV tracking system on tropic regions*. [Thesis]. Faculty of Engineering, University of Cundinamarca, Colombia, 2015.
- [14] M. Sidek, W. Hasan, M. Kadir, S. Shafie, M. Radzi, S. Ahmad e M. Marhaban, Artists, *GPS Based Portable Dual-Axis Solar Tracking System Using Astronomical Equation*. [Thesis]. Universiti Putra Malaysia, 2014.
- [15] D. Johnson-Hoyte, M. L. S. How, M. Thaw e D. Rossi, Artists, *Dual-Axis Solar Tracker: Functional Model Realization and Full-Scale Simulations*. [Thesis]. Faculty of Worcester Polytechnic Institute, 2013.
- [16] Catsquisher, "Wikimedia Commons," 23 Janeiro 2011. [Online]. Available: https://commons.wikimedia.org/wiki/File:Worm_Gear.gif. [Acedido em 23 Julho 2016].
- [17] K. Nice, "How Stuff Works," [Online]. Available: <http://science.howstuffworks.com/transport/engines-equipment/gear2.htm>. [Acedido em 23 Julho 2016].
- [18] K. Zipp, Dunkermotor e A. P. M. Control, "Solar Power World," 3 Abril 2013 . [Online]. Available: <http://www.solarpowerworldonline.com/2013/04/what-is-the-best-motor-for-solar-trackers/>. [Acedido em 10 Abril 2016].

- [19] “ASCEND,” 7 Novembro 2012. [Online]. Available: http://ascend4.org/Calculation_of_sun_position. [Acedido em 13 Abril 2016].
- [20] Trinamic Motion Control, “PDx-109-57 V2,” em *Hardware Manual*, Hamburg, Trinamic, 2010, pp. 14-16.
- [21] T. M. C. G. & C. KG, TMCL Reference and Programming Manual Version: 2.22, Hamburg: Trinamic, 2008.
- [22] J. Sanchez, “Home Power,” Maio 2010. [Online]. Available: <http://www.homepower.com/circuit-methods-testing>. [Acedido em 24 Julho 2016].
- [23] J. Baum, S. Johnson, W. Surles, A. Watrous e M. S. Zarske , “TeachEngineering digital library,” Regents of the University of Colorado, 2009 . [Online]. Available: https://www.teachengineering.org/lessons/view/cub_pveff_lesson03. [Acedido em 20 Novembro 2016].
- [24] A. E. Tutorials, “Alternative Energy Tutorials,” [Online]. Available: <http://www.alternative-energy-tutorials.com/energy-articles/solar-cell-i-v-characteristic.html>. [Acedido em 20 Novembro 2016].
- [25] PVGIS, “Photovoltaic Geographical Information System,” [Online]. Available: <http://re.jrc.ec.europa.eu/pvgis/apps4/pvest.php?lang=en&map=europe>. [Acedido em 26 Julho 2016].
- [26] “SunCalc,” [Online]. Available: <http://suncalc.net/#/38.6601,-9.2048,19/2016.07.31/00:17>. [Acedido em 31 Julho 2016].
- [27] C. Honsberg e S. Bowden, “PV Education,” [Online]. Available: <http://www.pveducation.org/pvcdrom/suns-position-high-accuracy>. [Acedido em 28 Julho 2016].
- [28] S. Rahman, “Green power: what is it and where can we find it?,” *IEEE Power and Energy Magazine*, vol. 1, pp. 30-37, 2003.
- [29] S. R. Bull, “Renewable energy today and tomorrow,” *Proceedings of the IEEE*, vol. 89, pp. 1216-1226, 2001.
- [30] S. J. Hamilton, Artist, *Sun-Tracking Solar Cell*. [Thesis]. University of Queensland, 1999.
- [31] K. S. O. Filho e M. F. O. Saraiva, “Astronomia e Astrofísica,” Instituto de Física da UFRGS, 28 Março 2012. [Online]. Available: <http://astro.if.ufrgs.br/tempo/mas.htm>. [Acedido em 10 Março 2016].

- [32] Zomeworks Corporation, "Zomeworks," Fevereiro 2012. [Online]. Available: <http://www.zomeworks.com/photovoltaic-tracking-racks/how-trackers-work>. [Acedido em 11 Março 2016].
- [33] S. Majumdar, "Homemade Circuit Projects," 24 Fevereiro 2016. [Online]. Available: <http://www.homemade-circuits.com/2016/02/ltr-light-dependent-resistors.html>. [Acedido em 24 Março 2016].
- [34] "ToolsForGreenLiving," 2016. [Online]. Available: <http://www.toolsforgreenliving.com/solar-trackers-technology>. [Acedido em 28 Março 2016].
- [35] A. Fong e J. Tippet, "Direction or angle of incidence," em *Project Development in the Solar Industry*, Boca Raton, CRC Press, 2012, pp. 123-127.
- [36] "The Solar Electricity Handbook," Greenstream Publishing Limited, 2016. [Online]. Available: <http://solarelectricityhandbook.com/solar-angle-calculator.html>. [Acedido em 20 Abril 2016].
- [37] M. A. Maehlum, "Energy Informative," 14 Agosto 2013. [Online]. Available: <http://energyinformative.org/solar-panel-tracking-systems>. [Acedido em 21 Julho 2016].
- [38] REUK, "REUK," 24 Setembro 2014. [Online]. Available: <http://www.reuk.co.uk/Measuring-the-Power-of-A-Solar-Panel.htm>. [Acedido em 24 Julho 2016].
- [39] A. Renewables, "AllEarth Renewables," [Online]. Available: <https://www.allearthrenewables.com/product/solar-tracking-systems-faq>. [Acedido em 3 Agosto 2016].

Anexos

Anexo 1: Medições e cálculos efetuados

Tabela I - Resultados do seguidor a funcionar com sensores LDR e com o algoritmo PSA

Hora	Medições	Sensores LDR			Algoritmo PSA			Fixo 34° Sul
		Dois Eixos	Elevação	Azimute	Dois Eixos	Elevação	Azimute	
10:00	V [V]	18,88	18,29	18,69	18,51	17,73	18,36	16,81
	I [A]	0,71	0,69	0,71	0,70	0,67	0,69	0,63
	P [W]	13,40	12,62	13,27	12,96	11,88	12,67	10,59
11:00	V [V]	18,78	18,34	18,33	18,29	17,99	18,03	17,44
	I [A]	0,72	0,70	0,70	0,70	0,69	0,69	0,67
	P [W]	13,52	12,84	12,83	12,80	12,41	12,44	11,68
12:00	V [V]	19,11	18,97	18,49	18,49	18,26	18,04	17,74
	I [A]	0,73	0,72	0,70	0,70	0,69	0,69	0,67
	P [W]	13,95	13,66	12,94	12,94	12,60	12,45	11,89
13:00	V [V]	19,10	18,80	17,96	18,40	18,30	17,70	17,70
	I [A]	0,72	0,71	0,68	0,69	0,69	0,67	0,67
	P [W]	13,75	13,35	12,21	12,70	12,63	11,86	11,86
14:00	V [V]	18,70	18,60	18,10	18,40	18,30	17,90	17,80
	I [A]	0,71	0,70	0,68	0,70	0,70	0,68	0,68
	P [W]	13,28	13,02	12,31	12,88	12,81	12,17	12,10
15:00	V [V]	19,10	18,90	18,50	18,70	18,50	18,30	18,00
	I [A]	0,73	0,72	0,70	0,71	0,71	0,70	0,68
	P [W]	13,94	13,61	12,95	13,28	13,14	12,81	12,24
16:00	V [V]	18,60	17,11	18,50	18,60	17,8	18,40	17,60
	I [A]	0,71	0,71	0,71	0,70	0,67	0,70	0,67
	P [W]	13,21	12,15	13,14	13,02	11,93	12,88	11,79
17:00	V [V]	18,60	16,73	18,50	18,60	16,80	18,60	16,70
	I [A]	0,71	0,68	0,71	0,71	0,64	0,71	0,64
	P [W]	13,21	11,38	13,14	13,21	10,75	13,21	10,69

Anexo 2: Código do PD3-109-57 (Motor 1)

```
amax = 200                                //max. acceleration (0 ... 2047)
vmax = 50                                  //max. speed (0 ... 2047)
cmax = 230                                 //max. current (0 ... 255)

// *** Initialization / Set up axis parameter *** //
SAP 5, 0, amax                            //set max. acceleration
SAP 6, 0, cmax                            //set max. current (0 ... 255)
SAP 12, 0, 0                             //enable right stop switch
SAP 13, 0, 0                             //enable left stop switch

SGP 80, 0, 1                             //enable shutdown configuration

// *** Main Loop *** //
loop:
    GIO 1, 0                             //read IN_0
    COMP 0                                //compare with 0
    JC NE, stopMotor                     //if not equal stop Motor

    GIO 2, 0
    COMP 0
    JC EQ, turnRight                     //if equal jump to turn right

    ROL 0, vmax                          //if not equal turn left at vmax
    JA loop                              //jump to loop

turnRight:
    ROR 0, vmax                          //turn right at vmax
    JA loop                              //jump to loop

stopMotor:
    MST 0
    JA loop
```


Anexo 3: Código do PD3-109-57 (Motor 2)

```
amax = 200                                //max. acceleration (0 ... 2047)
vmax = 50                                 //max. speed (0 ... 2047)
cmax = 100                                //max. current (0 ... 255)

// *** Initialization / Set up axis parameter *** //
SAP 5, 0, amax                            //set max. acceleration
SAP 6, 0, cmax                            //set max. current (0 ... 255)
SAP 12, 0, 0                             //enable right stop switch
SAP 13, 0, 0                             //enable left stop switch

SGP 80, 0, 1                             //enable shutdown configuration

// *** Main Loop *** //
loop:
    GIO 1, 0                             //read IN_0
    COMP 0                                //compare with 0
    JC NE, stopMotor                     //if not equal stop Motor

    GIO 2, 0
    COMP 0
    JC EQ, turnRight                     //if equal jump to turn right

    ROL 0, vmax                          //if not equal turn left at vmax
    JA loop                               //jump to loop

turnRight:
    ROR 0, vmax                          //turn right at vmax
    JA loop                               //jump to loop

stopMotor:
    MST 0
    JA loop
```

Anexo 4: Código do Arduino (Seguidor com sensores LDR)

```
#include <avr/io.h>
#include <avr/interrupt.h>

//-----Set variables-----
int tracking_Sensibility = 30;           //Sensibility to trigger rotation
int delay_magnetic_Brake = 800;         //1000 = 1 second

boolean activateNightMode = false;
boolean trackElevation = true;
boolean trackAzimuth = true;

const int LDR_Night_Detection = 300;    //Sensibility to detect night operation

const int timer_limit = 10;              //Time in seconds to trigger the Interrupt Service Routine
volatile byte seconds = 0;               //seconds on timer, start counting from 0
//-----

const int pin_LDR1 = 0;                  //LDR1 (left) = analog pin 0           (input)
const int pin_LDR2 = 1;                  //LDR2 (right) = analog pin 1          (input)
const int pin_LDR3 = 2;                  //LDR2 (right) = analog pin 2          (input)
const int pin_LDR4 = 3;                  //LDR2 (right) = analog pin 3          (input)

const int pin_limit_Switch_Top = 2;       //Top Limit Switch = digital pin 2      (input)
const int pin_limit_Switch_Down = 3;      //Down Limit Switch = digital pin 3     (input)
const int pin_stop_Motor_1 = 4;           //Stop Motor1 (elevation) = digital pin 4 (output)
const int pin_direction_Motor_1 = 5;      //Direction of Motor1 (elevation) = digital pin 5 (output)
const int pin_shutdown_routine = 6;       //Enter shutdown routine = digital pin 6 (input)
const int pin_shutdown_Motor_1 = 7;       //Stepper Motor1 Shutdown = digital pin 7 (output)

const int pin_limit_Switch_Right = 8;     //Right Limit Switch = digital pin 8    (input)
const int pin_limit_Switch_Left = 9;      //Left Limit Switch = digital pin 9     (input)
const int pin_stop_Motor_2 = 10;          //Stop Motor1 (elevation) = digital pin 10 (output)
const int pin_direction_Motor_2 = 11;     //Direction of Motor2 (azimuth) = digital pin 11 (output)
const int pin_magnetic_Brake_2 = 12;      //Magnetic Brake2 = digital pin 12      (output)
const int pin_shutdown_Motor_2 = 13;      //Stepper Motor2 Shutdown = digital pin 13 (output)
```

```

int get_LDR1;                //analog value of LDR1
int get_LDR2;                //analog value of LDR2
int get_LDR3;                //analog value of LDR3
int get_LDR4;                //analog value of LDR4

int diff_LDR1_LDR2;          //difference between LDR1 and LDR2
int diff_LDR3_LDR4;          //difference between LDR3 and LDR4

boolean get_limit_Switch_Top;
boolean get_limit_Switch_Down;
boolean get_limit_Switch_Left;
boolean get_limit_Switch_Right;

boolean shutdown_routine;    //store state of pin_shutdown_routine

void setup()
{
    Serial.begin(9600);

    //set output pins
    pinMode(pin_stop_Motor_1, OUTPUT);
    pinMode(pin_direction_Motor_1, OUTPUT);
    pinMode(pin_shutdown_routine, INPUT);
    pinMode(pin_shutdown_Motor_1, OUTPUT);

    pinMode(pin_stop_Motor_2, OUTPUT);
    pinMode(pin_direction_Motor_2, OUTPUT);
    pinMode(pin_magnetic_Brake_2, OUTPUT);
    pinMode(pin_shutdown_Motor_2, OUTPUT);

    //set input pins
    pinMode(pin_limit_Switch_Top, INPUT);
    pinMode(pin_limit_Switch_Down, INPUT);
    pinMode(pin_limit_Switch_Left, INPUT);
    pinMode(pin_limit_Switch_Right, INPUT);

    // initialize Timer1:
    cli();          // disable global interrupts
    TCCR1A = 0;     // set entire TCCR1A register to 0
    TCCR1B = 0;     // same for TCCR1B

```

```

// set compare match register to desired timer count:
OCR1A = 15624;

// turn on CTC mode:
TCCR1B |= (1 << WGM12);

// Set CS10 and CS12 bits for 1024 prescaler:
TCCR1B |= (1 << CS10);
TCCR1B |= (1 << CS12);

// enable timer compare interrupt:
TIMSK1 |= (1 << OCIE1A);

// enable global interrupts:
sei();
}

//Interrupt Service Routine (set to run per second)
ISR(TIMER1_COMPA_vect)
{
    seconds++;

    if(seconds == timer_limit)
    {
        seconds = 0;      //reset sleep timer of Stepper Motor1

        digitalWrite(pin_shutdown_Motor_2, LOW);      //set Stepper Motor2 to sleep
    }
}

void loop()
{
    shutdown_routine = digitalRead(pin_shutdown_routine);

    get_limit_Switch_Top = digitalRead(pin_limit_Switch_Top);
    get_limit_Switch_Down = digitalRead(pin_limit_Switch_Down);
    get_limit_Switch_Left = digitalRead(pin_limit_Switch_Left);
    get_limit_Switch_Right = digitalRead(pin_limit_Switch_Right);
}

```

```

get_LDR1 = analogRead(pin_LDR1);           //get analog value of LDR1 (bottom right: orange)
get_LDR2 = analogRead(pin_LDR2);           //get analog value of LDR2 (top left: brown)
get_LDR3 = analogRead(pin_LDR3);           //get analog value of LDR3 (bottom left: green)
get_LDR4 = analogRead(pin_LDR4);           //get analog value of LDR4 (top right: blue)

//Day Night Tracking
if ( (shutdown_routine == true) || ((activateNightMode == true) && (get_LDR1 < LDR_Night_Detection)
&& (get_LDR2 < LDR_Night_Detection) && (get_LDR3 < LDR_Night_Detection) && (get_LDR4 <
LDR_Night_Detection)) )
{
  Serial.println();
  Serial.println("*** Night Mode ***");
  //Rotate Top
  if(get_limit_Switch_Top==LOW)
  {
    Serial.println("Motor 1: Rotate Up!!");
    digitalWrite(pin_shutdown_Motor_1, HIGH); //wake up Stepper Motor1
    digitalWrite(pin_direction_Motor_1, HIGH); //set motor1 direction: right
    digitalWrite(pin_stop_Motor_1, HIGH);      //enable motor1 rotation
  }
  else
  {
    Serial.println("Top Limit Switch Activated => Shutdown Motor 1");

    //Shutdown Motor 1
    delay(5000);
    digitalWrite(pin_shutdown_Motor_1, LOW);      //set Stepper Motor1 to sleep

    //Shutdown Motor 2
    digitalWrite(pin_stop_Motor_2, LOW);           //stop motor2 rotation
    delay(delay_magnetic_Brake);                   //delay
    digitalWrite(pin_magnetic_Brake_2, LOW);        //lock system with DC Solenoid
    digitalWrite(pin_shutdown_Motor_2, LOW);        //set Stepper Motor2 to sleep
  }
}
else
{
  Serial.println();

  //Track Elevation

```

```

    if (trackElevation==true)
    {
        motor1_Tracking();
    }

    //Track Azimuth
    if (trackAzimuth==true)
    {
        motor2_Tracking();
    }
}

//-----
//Motor 1
//-----
void motor1_Tracking()
{
    diff_LDR1_LDR2 = abs(get_LDR1 - get_LDR2);           //get the absolute value of the subtraction between
                                                         //LDR1 and LDR2

    //Stall Motor
    //if the gap of LDR1 and LDR2 is within the Sensibility Range, then LDR1 and LDR2 are equal
    //if the Sun is outside of tracking range in Elevation and Limit Switch is activated, then stop motor 1 and
    //lock DC Solenoid
    if ((diff_LDR1_LDR2 <= tracking_Sensibility) || ((get_LDR1 < get_LDR2) &&
(get_limit_Switch_Top==HIGH)) || ((get_LDR1 > get_LDR2) && (get_limit_Switch_Down==HIGH)))
    {
        Serial.println("Stall: Motor 1");
        digitalWrite(pin_stop_Motor_1, LOW);             //stop motor1 rotation
    }

    //Up Movement: Motor1 Rotate Right
    else if (get_LDR1 < get_LDR2)
    {
        Serial.println("Motor 1: Rotate Up!!");
        digitalWrite(pin_shutdown_Motor_1, HIGH);         //wake up Stepper Motor1
        digitalWrite(pin_direction_Motor_1, HIGH);         //(set motor1 direction: right)
        digitalWrite(pin_stop_Motor_1, HIGH);             //enable motor1 rotation
    }

    //Down Movement: Motor1 Rotate Left
    else

```

```

{
    Serial.println("Motor 1: Rotate Down!!");
    digitalWrite(pin_shutdown_Motor_1, HIGH);           //wake up Stepper Motor1
    digitalWrite(pin_direction_Motor_1, LOW);           //(set motor1 direction: left)
    digitalWrite(pin_stop_Motor_1, HIGH);               //enable motor1 rotation
}
}
//-----
//Motor 2
//-----
void motor2_Tracking()
{
    diff_LDR3_LDR4 = abs(get_LDR3 - get_LDR4);          //get the absolute value of the subtraction between
                                                         //LDR3 and LDR4

    //Stall Motor
    //if the gap of LDR3 and LDR4 is within the Sensibility Range, then LDR3 and LDR4 are equal
    //if the Sun is outside of tracking range in Azimuth and Limit Switch is activated, then stop motor 2 and
    //lock DC Solenoid
    if ((diff_LDR3_LDR4 <= tracking_Sensibility) || ((get_LDR3 < get_LDR4) &&
(get_limit_Switch_Right==HIGH)) || ((get_LDR3 > get_LDR4) && (get_limit_Switch_Left==HIGH)))
    {
        Serial.println("Stall: Motor 2");
        digitalWrite(pin_stop_Motor_2, LOW);            //stop motor2 rotation

        delay(delay_magnetic_Brake);                   //delay
        digitalWrite(pin_magnetic_Brake_2, LOW);        //lock system with DC Solenoid
    }
    //Right Movement
    else if (get_LDR3 < get_LDR4)
    {
        Serial.println("Motor 2: Rotate Right!!");
        digitalWrite(pin_magnetic_Brake_2, HIGH);       //unlock system with DC Solenoid
        delay(delay_magnetic_Brake);                   //delay

        digitalWrite(pin_shutdown_Motor_2, HIGH);       //wake up Stepper Motor2
        digitalWrite(pin_direction_Motor_2, HIGH);      //set motor1 direction: right
        digitalWrite(pin_stop_Motor_2, HIGH);           //enable motor2 rotation
        seconds = 0;                                    //reset sleep timer of Stepper Motor2
    }
    //Left Movement

```

```

else
{
  Serial.println("Motor 2: Rotate Left!!");
  digitalWrite(pin_magnetic_Brake_2, HIGH);           //unlock system with DC Solenoid
  delay(delay_magnetic_Brake);                       //delay

  digitalWrite(pin_shutdown_Motor_2, HIGH);           //wake up Stepper Motor2
  digitalWrite(pin_direction_Motor_2, LOW);           //set motor2 direction: left
  digitalWrite(pin_stop_Motor_2, HIGH);              //enable motor2 rotation
  seconds = 0;                                       //reset sleep timer of Stepper Motor2
}
}

```


Anexo 5: Código do Arduino (Seguidor com algoritmo PSA)

```
//(C) 2011 Hannes Hassler.
//updated and checked 2015
#include <avr/io.h>
#include <avr/interrupt.h>

#include "Wire.h"
#include <Helios.h>
#include <DS1107H.h>

Helios helios;
DS1107H clock;

//-----Set variables-----
double latitude = 38.660073;
double longitude = -9.204856;

float currentAngle_Elevation=85.5;           //current angle of solar collector
float currentAngle_Azimuth=107.2;           //current angle of solar collector
float refenceAngle_1 = 2.7;                 //2.7° per 1 second of rotation (velocity of rotation)
float refenceAngle_2 = 2.8;                 //2.8° per 1 second of rotation (velocity of rotation)

boolean trackElevation = true;
boolean trackAzimuth = true;

const float tracking_Sensibility = 1;        //Sensibility to trigger rotation in degrees

int delay_magnetic_Brake = 0;               //1000 = 1 second

const int timer_limit = 60;                 //Time in seconds to trigger the Interrupt Service Routine
volatile byte seconds = 0;                  //seconds on timer, start counting from 0
//-----

const int pin_limit_Switch_Top = 2;          //Top Limit Switch = digital pin 2           (input)
const int pin_limit_Switch_Down = 3;         //Down Limit Switch = digital pin 3           (input)
const int pin_stop_Motor_1 = 4;              //Stop Motor1 (elevation) = digital pin 4      (output)
const int pin_direction_Motor_1 = 5;         //Direction of Motor1 (elevation) = digital pin 5 (output)
const int pin_shutdown_routine = 6;          //Enter shutdown routine = digital pin 6       (input)
const int pin_shutdown_Motor_1 = 7;          //Stepper Motor1 Shutdown = digital pin 7      (output)
```

```

const int pin_limit_Switch_Right = 8;           //Right Limit Switch = digital pin 8           (input)
const int pin_limit_Switch_Left = 9;           //Left Limit Switch = digital pin 9           (input)
const int pin_stop_Motor_2 = 10;               //Stop Motor1 (elevation) = digital pin 10     (output)
const int pin_direction_Motor_2 = 11;          //Direction of Motor2 (azimuth) = digital pin 11 (output)
const int pin_magnetic_Brake_2 = 12;           //Magnetic Brake2 = digital pin 12             (output)
const int pin_shutdown_Motor_2 = 13;           //Stepper Motor2 Shutdown = digital pin 13     (output)
int startTracking;
boolean get_limit_Switch_Top;
boolean get_limit_Switch_Down;
boolean get_limit_Switch_Left;
boolean get_limit_Switch_Right;
boolean shutdown_routine;                     //store state of pin_shutdown_routine
boolean initialization=true;                   //set solar tracker to the left limit
float rotationLength;                         //time in seconds of motor rotation to reach the actual
                                              //angle of the sun's position
float newAngle;                               //angle of solar collector after reaching the sun position
double dAzimuth;
double dElevation;

void setup()
{
  Serial.begin(9600);

  //set output pins
  pinMode(pin_stop_Motor_1, OUTPUT);
  pinMode(pin_direction_Motor_1, OUTPUT);
  pinMode(pin_shutdown_routine, INPUT);
  pinMode(pin_shutdown_Motor_1, OUTPUT);

  pinMode(pin_stop_Motor_2, OUTPUT);
  pinMode(pin_direction_Motor_2, OUTPUT);
  pinMode(pin_magnetic_Brake_2, OUTPUT);
  pinMode(pin_shutdown_Motor_2, OUTPUT);

  //uncomment following to set the timer chip
  //Once done, don't forget to comment out again, otherwise
  //you get the same time with each startup ;-)

  /*

```

```

byte second=0;
byte minute=27;
byte hour=14;
byte dayOfWeek=2;
byte dayOfMonth=12;
byte month=7;
byte year=16;
clock.setDate(second,minute,hour,dayOfWeek,dayOfMonth,month,year);
*/
//-----
// initialize Timer1:
//-----
cli();           // disable global interrupts
TCCR1A = 0;      // set entire TCCR1A register to 0
TCCR1B = 0;      // same for TCCR1B

// set compare match register to desired timer count:
OCR1A = 15624;

// turn on CTC mode:
TCCR1B |= (1 << WGM12);

// Set CS10 and CS12 bits for 1024 prescaler:
TCCR1B |= (1 << CS10);
TCCR1B |= (1 << CS12);

// enable timer compare interrupt:
TIMSK1 |= (1 << OCIE1A);

// enable global interrupts:
sei();
//-----
}

//Interrupt Service Routine (set to run per second)
ISR(TIMER1_COMPA_vect)
{
  seconds++;
  if(seconds == timer_limit)
  {
    seconds = 0;
    //reset sleep timer of Stepper Motor1
  }
}

```

```

        digitalWrite(pin_shutdown_Motor_2, LOW);          //set Stepper Motor2 to sleep
    }
}
void loop()
{
    Serial.println();
    clock.getDate();
    helios.calcSunPos(clock.year, clock.month, clock.dayOfMonth,
    clock.hour, clock.minute, clock.second, longitude, latitude);
    showTime(clock);
    shutdown_routine = digitalRead(pin_shutdown_routine);
    get_limit_Switch_Top = digitalRead(pin_limit_Switch_Top);
    get_limit_Switch_Down = digitalRead(pin_limit_Switch_Down);
    get_limit_Switch_Left = digitalRead(pin_limit_Switch_Left);
    get_limit_Switch_Right = digitalRead(pin_limit_Switch_Right);
    //Shutdown Routine -> Rotate Top
    if(shutdown_routine==true)
    {
        //Rotate Top
        if(get_limit_Switch_Top==LOW)
        {
            Serial.println("Shutting Down Motor 1: Rotate Top!!");
            digitalWrite(pin_shutdown_Motor_1, HIGH);          //wake up Stepper Motor1
            digitalWrite(pin_direction_Motor_1, HIGH);          //set motor1 direction: right
            digitalWrite(pin_stop_Motor_1, HIGH);               //enable motor1 rotation
        }
        else
        {
            //Shutdown Motor 1
            Serial.println("Top Limit Switch Activated => Shutdown Motor 1");
            delay(5000);
            digitalWrite(pin_shutdown_Motor_1, LOW);            //set Stepper Motor1 to sleep
            //Shutdown Motor 2
            Serial.println("Shutdown Motor 2");
            digitalWrite(pin_stop_Motor_2, LOW);                //stop motor2 rotation
            delay(delay_magnetic_Brake);                        //delay
            digitalWrite(pin_magnetic_Brake_2, LOW);             //lock system with DC Solenoid
            digitalWrite(pin_shutdown_Motor_2, LOW);             //set Stepper Motor2 to sleep
        }
    }
}

```

```

else
{
    //Track Azimuth
    if(trackAzimuth==true)
    {
        motor2_Tracking();
    }
    //Tack Elevation
    if(trackElevation==true)
    {
        motor1_Tracking();
    }
}
//-----
//Motor 1
//-----
void motor1_Tracking()
{
    dElevation=helios.dElevation;show(">Sun Elevation",dElevation,true);
    Serial.print("#Solar tracker elevation: ");
    Serial.println(currentAngle_Elevation);

    newAngle = abs(dElevation - currentAngle_Elevation);           //degrees between the solar tracker and
                                                                    //the sun position
    rotationLength = newAngle*1000L/refenceAngle_1;                 //time in seconds for motor to rotate
    if(dElevation > (currentAngle_Elevation+tracking_Sensibility))
    {
        Serial.print("#Seconds of motor rotation: ");
        Serial.println(rotationLength/1000L);
        digitalWrite(pin_shutdown_Motor_1, HIGH);                 //wake up Stepper Motor1
        digitalWrite(pin_direction_Motor_1, HIGH);                 //set motor1 direction: right
        digitalWrite(pin_stop_Motor_1, HIGH);                      //enable motor1 rotation
        Serial.println("#Rotate Up!!!");
        delay(rotationLength);
        digitalWrite(pin_stop_Motor_1, LOW);                       //disable motor1 rotation
        currentAngle_Elevation=dElevation;
    }
    else if(dElevation < (currentAngle_Elevation-tracking_Sensibility))
    {

```

```

Serial.print("#Seconds of motor rotation: ");
Serial.println(rotationLength/1000L);
digitalWrite(pin_shutdown_Motor_1, HIGH);           //wake up Stepper Motor1
digitalWrite(pin_direction_Motor_1, LOW);           //set motor1 direction: left
digitalWrite(pin_stop_Motor_1, HIGH);               //enable motor1 rotation
Serial.println("#Rotate Down!!!");
    delay(rotationLength);
    digitalWrite(pin_stop_Motor_1, LOW);             //disable motor1 rotation
    currentAngle_Elevation=dElevation;
}
}
//-----
//Motor 2
//-----
void motor2_Tracking()
{
    Serial.print("get_limit_Switch_Left == ");
    Serial.println(get_limit_Switch_Left);
    Serial.print("get_limit_Switch_Right == ");
    Serial.println(get_limit_Switch_Right);

    //Initialize solar tracker, set it to left limit position
    if(get_limit_Switch_Right == LOW && initialization==true)
    {
        Serial.println("Motor 2: Initialization Rotate Right!!");
        digitalWrite(pin_magnetic_Brake_2, HIGH);    //unlock system with DC Solenoid
        delay(delay_magnetic_Brake);                 //delay

        digitalWrite(pin_shutdown_Motor_2, HIGH);    //wake up Stepper Motor2
        digitalWrite(pin_direction_Motor_2, HIGH);    //set motor2 direction: right
        digitalWrite(pin_stop_Motor_2, HIGH);         //enable motor2 rotation
        seconds = 0;                                  //reset sleep timer of Stepper Motor2
    }
    else
    {
        if (get_limit_Switch_Right == HIGH)
        digitalWrite(pin_magnetic_Brake_2, LOW);     //unlock system with DC Solenoid
        delay(5000);                                  //5 seconds to stabilize movement
                                                //oscillations when motor stop
    }
}

```

```

dAzimuth=helios.dAzimuth;show(">Sun Azimuth",dAzimuth,true);
Serial.print("#Solar tracker azimuth: ");
Serial.println(currentAngle_Azimuth);
newAngle = abs(dAzimuth - currentAngle_Azimuth);           //degrees between the solar tracker and
                                                             //the sun position
rotationLength = newAngle*1000L/refenceAngle_2;             //time in seconds for motor to rotate
if(dAzimuth > (currentAngle_Azimuth+tracking_Sensibility))
{
  Serial.print("#Seconds of motor rotation: ");
  Serial.println(rotationLength/1000L);
  digitalWrite(pin_magnetic_Brake_2, HIGH);                 //unlock system with DC Solenoid
  delay(delay_magnetic_Brake);                               //delay
  digitalWrite(pin_shutdown_Motor_2, HIGH);                 //wake up Stepper Motor2
  digitalWrite(pin_direction_Motor_2, LOW);                 //set motor2 direction: left
  digitalWrite(pin_stop_Motor_2, HIGH);                     //enable motor2 rotation
  Serial.println("#Rotate Left!!!");
  delay(rotationLength);
  digitalWrite(pin_stop_Motor_2, LOW);                       //disable motor2 rotation
  delay(delay_magnetic_Brake);                               //delay
  digitalWrite(pin_magnetic_Brake_2, LOW);                 //unlock system with DC Solenoid
  currentAngle_Azimuth=dAzimuth;
  seconds = 0;                                               //reset sleep timer of Stepper Motor2
}
else if(dAzimuth < (currentAngle_Azimuth-tracking_Sensibility))
{
  Serial.print("#Seconds of motor rotation: ");
  Serial.println(rotationLength/1000L);
  digitalWrite(pin_magnetic_Brake_2, HIGH);                 //unlock system with DC Solenoid
  delay(delay_magnetic_Brake);                               //delay
  digitalWrite(pin_shutdown_Motor_2, HIGH);                 //wake up Stepper Motor2
  digitalWrite(pin_direction_Motor_2, HIGH);                //set motor2 direction: right
  digitalWrite(pin_stop_Motor_2, HIGH);                     //enable motor2 rotation
  Serial.println("#Rotate Right!!!");
  delay(rotationLength);
  digitalWrite(pin_stop_Motor_2, LOW);                       //disable motor2 rotation
  delay(delay_magnetic_Brake);                               //delay
  digitalWrite(pin_magnetic_Brake_2, LOW);                 //unlock system with DC Solenoid
  currentAngle_Azimuth=dAzimuth;
  seconds = 0;                                               //reset sleep timer of Stepper Motor2
}

```

```

    initialization=false;
  }
}

void show(char nameStr[], double val, boolean newline) {
  Serial.print(nameStr);
  Serial.print("=");
  if (newline)
    Serial.println(val);
  else Serial.print(val);
}

void showTime(DS1107H timerChip) {
  Serial.print("UT ");
  Serial.print(timerChip.hour, DEC);
  Serial.print(":");
  Serial.print(timerChip.minute, DEC);
  Serial.print(":");
  Serial.print(timerChip.second, DEC);
  Serial.print(" ");
  Serial.print(timerChip.month, DEC);
  Serial.print("/");
  Serial.print(timerChip.dayOfMonth, DEC);
  Serial.print("/");
  Serial.print(timerChip.year, DEC);
  Serial.print(" Day_of_week:");
  Serial.println(timerChip.dayOfWeek, DEC);
}

```